

PCT

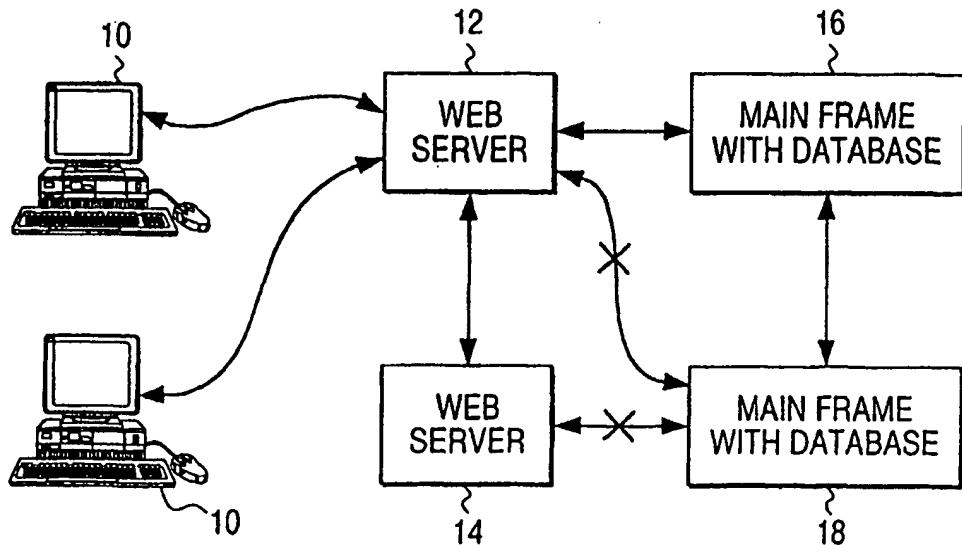
WORLD INTELLECTUAL PROPERTY ORGANIZATION
International Bureau



INTERNATIONAL APPLICATION PUBLISHED UNDER THE PATENT COOPERATION TREATY (PCT)

(51) International Patent Classification ⁷ : G06F 15/00, 17/30	A1	(11) International Publication Number: WO 00/33202 (43) International Publication Date: 8 June 2000 (08.06.00)
<p>(21) International Application Number: PCT/US99/28321</p> <p>(22) International Filing Date: 1 December 1999 (01.12.99)</p> <p>(30) Priority Data: 09/203,686 1 December 1998 (01.12.98) US</p> <p>(71) Applicant: UNIVERSITY OF FLORIDA [US/US]; 223 Grinter Hall, P.O. Box 115500, Gainesville, FL 32611-5500 (US).</p> <p>(72) Inventors: ROBBINS, James, Earl; University of Florida, P.O. Box 114000, Gainesville, FL 32611-4000 (US). COOK, Alan, Russell; University of Florida, P.O. Box 114000, Gainesville, FL 32611-4000 (US). LUCAS, Michael, Wayne; University of Florida, P.O. Box 114000, Gainesville, FL 32611-4000 (US).</p> <p>(74) Agent: CLARKE, Dennis, P.; Kerkam, Stowell, Kondracki & Clarke, P.C., Suite 600, Two Skyline Place, 5203 Leesburg Pike, Falls Church, VA 22041-3401 (US).</p>		<p>(81) Designated States: AE, AL, AU, BA, BB, BG, BR, CA, CN, CU, CZ, EE, GE, HR, HU, ID, IL, IN, IS, JP, KP, KR, LC, LK, LR, LT, LV, MG, MK, MN, MX, NO, NZ, PL, RO, SI, SK, SL, TR, TT, UA, UZ, VN, YU, ZA, ARIPO patent (GH, GM, KE, LS, MW, SD, SL, SZ, TZ, UG, ZW), Eurasian patent (AM, AZ, BY, KG, KZ, MD, RU, TJ, TM), European patent (AT, BE, CH, CY, DE, DK, ES, FI, FR, GB, GR, IE, IT, LU, MC, NL, PT, SE), OAPI patent (BF, BJ, CF, CG, CI, CM, GA, GN, GW, ML, MR, NE, SN, TD, TG).</p> <p>Published <i>With international search report. Before the expiration of the time limit for amending the claims and to be republished in the event of the receipt of amendments.</i></p>

(54) Title: **WEB PAGE ACCESSING OF DATA BASES AND MAINFRAMES**



(57) Abstract

A method of providing Web access to data using dynamic generation of Web pages by a mainframe computer (16, 18) connected to a Web server (12, 14). The mainframe computer (16, 18) has legacy programs, legacy data, and legacy subroutines on it (legacy means existing information systems which may not be specifically adapted to Web access). A Web control program on the mainframe generates Web pages for the user. A state key is generated by the mainframe upon a user inputting identification data and is used to determine access allowed without the user reentering identification data.

FOR THE PURPOSES OF INFORMATION ONLY

Codes used to identify States party to the PCT on the front pages of pamphlets publishing international applications under the PCT.

AL	Albania	ES	Spain	LS	Lesotho	SI	Slovenia
AM	Armenia	FI	Finland	LT	Lithuania	SK	Slovakia
AT	Austria	FR	France	LU	Luxembourg	SN	Senegal
AU	Australia	GA	Gabon	LV	Latvia	SZ	Swaziland
AZ	Azerbaijan	GB	United Kingdom	MC	Monaco	TD	Chad
BA	Bosnia and Herzegovina	GE	Georgia	MD	Republic of Moldova	TG	Togo
BB	Barbados	GH	Ghana	MG	Madagascar	TJ	Tajikistan
BE	Belgium	GN	Guinea	MK	The former Yugoslav Republic of Macedonia	TM	Turkmenistan
BF	Burkina Faso	GR	Greece	ML	Mali	TR	Turkey
BG	Bulgaria	HU	Hungary	MN	Mongolia	TT	Trinidad and Tobago
BJ	Benin	IE	Ireland	MR	Mauritania	UA	Ukraine
BR	Brazil	IL	Israel	MW	Malawi	UG	Uganda
BY	Belarus	IS	Iceland	MX	Mexico	US	United States of America
CA	Canada	IT	Italy	NE	Niger	UZ	Uzbekistan
CF	Central African Republic	JP	Japan	NL	Netherlands	VN	Viet Nam
CG	Congo	KE	Kenya	NO	Norway	YU	Yugoslavia
CH	Switzerland	KG	Kyrgyzstan	NZ	New Zealand	ZW	Zimbabwe
CI	Côte d'Ivoire	KP	Democratic People's Republic of Korea	PL	Poland		
CM	Cameroon	KR	Republic of Korea	PT	Portugal		
CN	China	KZ	Kazakhstan	RO	Romania		
CU	Cuba	LC	Saint Lucia	RU	Russian Federation		
CZ	Czech Republic	LI	Liechtenstein	SD	Sudan		
DE	Germany	LK	Sri Lanka	SE	Sweden		
DK	Denmark	LR	Liberia	SG	Singapore		
EE	Estonia						

WEB PAGE ACCESSING OF DATA BASES AND MAINFRAMES

Copyright

A portion of the disclosure of this patent document contains material which is subject to copyright protection. The copyright owner has no objection to the facsimile reproduction by any one of the patent disclosure, as it appears in the Patent and Trademark office patent files or records, but otherwise reserves all copyright rights whatsoever.

Background of the Invention

The present invention relates to distribution of information over a computer network using hypertext links and Web pages. More specifically, the present invention relates to information distribution by interaction of a Web page on a client, a Web server connected to the client via the Internet, and programs, such as a database program, on a mainframe computer.

The explosive growth of the Internet over the last several years has dramatically improved the ability to quickly distribute vast amounts of data to any interested person. Additionally, the use of cookies, Secure Sockets Layers (SSL), and numerous other techniques have allowed the Internet to provide distribution of information restricted to an authorized person only.

An example may best illustrate some considerations in the situation. A book store may have a Web server that allows customers and potential customers access to information. The information might be "unrestricted

access" information such as a list of books that the book store is willing to sell. The information might be "customer specific" information, such as the list of books currently under order by a customer, the customer's account balance, or other such information that should be accessible only to the given customer (apart from appropriate employees of the book store who may need access to such restricted information). Moreover, there might be information available to a class of customers, but which should not be accessible by all persons. Thus, a list of books at special discounted prices could be available to any member of a preferred buyer group, but not to others.

Regardless of whether information is unrestricted, restricted to a single user, or has some intermediate level of restriction, distribution of the information via the Web requires that a Web server handle the information. Several methods exist to enable the Web server to handle the information.

The most direct technique is for the book store's programmers or other employees to put the information on the Web server. Thus, a book store setting up a Web site can simply take its list of books from whatever computer it is currently stored on and reformat the list as needed. However, when the list is updated at its current storage location (such as a mainframe computer), the book store's employees would also have to update the list on the Web server. The disadvantages of requiring the same information to be updated at multiple storage locations

(such as in the mainframe computer and in the Web server), the possibility of data updates being made inconsistently, and the increase in storage demands required when the data must be stored in one location (such as a mainframe) for internal company use and in another location (such as a Web server) for access via the internet or other outside access.

Another technique for providing Web access to data is for the book store's programmers to write specific interface programs to translate data stored in a database on the book store's mainframe computer. For example, an interface program can convert database data into hypertext markup language (HTML) such that a Web browser may display the data as a Web page. However, generating such interface programs requires skills that may be outside the experience of the employees who manage the mainframe and its databases. Further, such interface programs may be required for different types of data access. If a list of available books is stored in one database with a given format, and a list of customers is stored in a different format in a different database, a different interface program may be required for each. Considering also that a list of book publishers may be stored in yet another format and that the book store might allow each book publisher to view its own account information, yet another interface program would have to be developed. Developing interface programs for each database or each data format is expensive and time-consuming.

In addition to problems associated with allowing Web access to data on mainframe databases, security considerations often complicate access to information. For example, if the book store wanted to allow customers access to details about the status of their book orders or their account, the Web server often places a so-called cookie on the customer's computer. The cookie is a small file of data or code identifying the customer's computer to the Web server. However, many computer users do not want their computers accepting cookies from Web servers and they therefore limit the use of this. A further problem with some Web access security arrangements is that some are too easily overcome. For example, a person may "spoof" another's identity (impersonate another) in order to obtain restricted data.

Yet another consideration in distributing large amounts of data via the Internet is the need to be able to manage and administer a Web site. Often Web site management requires complex knowledge and considerable effort. Changing the appearance of a Web page (such as changing the presentation of data on a Web page) can be a time-consuming process which requires highly specialized expertise. (As used herein, changing the appearance means causing the Web page to appear differently to all who access it by causing it to display data from the computer storing the Web page, and shall not include simply appending user comments or designs to a Web page as is

common for Web-based bulletin board or public comment boards.)

Objects and Summary of the Invention

Accordingly, it is a primary object of the present invention to provide a new and improved technique of distributing data via the Internet or other computer network, bulletin board, etc.

A more specific object of the present invention is to provide for Internet distribution of data stored in a database without the need for storing the data on a Web server.

A further object of the present invention is to provide ready Web access to data stored in various formats without the need to develop different programs.

Yet another object of the present invention is to provide a new and improved security arrangement for limiting access to Web data to authorized users.

A further object of the present invention is to provide convenient management of a Web site without requiring advanced programming skills or expertise in HTML.

The above and other features of the present invention which will be more readily understood when the following detailed description is considered in conjunction with the accompanying drawings are realized by a method of providing Web access to data. A Web server for distribution of data to users is provided. A database is

provided on a database computer operably connected to the Web server, the database having a database program that provides data in a given format. Upon a user requesting data in the database, the Web server forwards the request to the database, the database program accesses the data in the given format. The database computer runs a Web control program that generates a Web page with the requested data and supplies the generated Web page to the user. The Web control program marries file definition objects and page definition objects to generate Web pages. The Web server is on a Web server computer different from the database computer. The database computer is a mainframe computer. The mainframe computer supplies the generated Web page to the user via the Web server computer. Upon a user requesting data not in the mainframe computer, the Web control program causes the mainframe computer to access data on a remote computer and the Web control program then generates a Web page with the requested data and supplies the generated Web page to the user. Responsive to a user, the Web control program calls a subroutine on the mainframe computer, which subroutine is independent of the Web control program and is a legacy subroutine. The method further includes the steps of: having a user supply an identification code, forwarding the identification code from the Web server computer to the mainframe computer, the Web control program then generating a state key, incorporating the state key into a Web page supplied to that user, and wherein the mainframe computer detects the

state key from the Web page upon the user requesting data and decides whether that user can access the requested data based on the state key. The state key is randomly generated for a given access session and the state key loses its ability to authorize data access if it is not sent by the user to the mainframe computer at least once during a time-out interval. The Web control program receives a page change command from a user using a Web browser and, responsive to the page change command, the Web control program changes a given Web page to a changed Web page such that any users accessing the given Web page now receive the changed Web page, the page change command changing the appearance of a given Web page.

The present invention may alternately be described as a method of providing Web access to data, the steps including: providing a Web server for distribution of data to users; providing a database on a database computer operably connected to the Web server, the database having a database program that provides data in a given format; providing a Web control program on the database computer; having a user supply an identification code to the Web server, the Web control program then generating a state key and incorporating the state key into a Web page supplied to that user, and wherein the database computer detects the state key from the Web page upon the user requesting data and decides whether that user can access the requested data based on the state key. The state key loses its ability to authorize data access if it is not

sent by the user to the database computer at least once during a time-out interval. The Web server is on a Web server computer different from the database computer. The method of further includes the step of forwarding the identification code from the Web server to the database computer. The database computer is a mainframe computer. The state key is randomly generated for a given access session.

The present invention may alternately be described as a method of managing a Web site, the steps including: providing a Web control program on a computer; having the Web control program receive a page change command from a user using a Web browser to access the Web site; and, responsive to the page change command, the Web control program changing a given Web page to a changed Web page such that any users accessing the given Web page now receive the changed Web page, the page change command changing the appearance of a given Web page. The Web control program marries file definition objects and page definition objects to generate Web pages.

The computer having the Web control program is a mainframe computer and users request data on the mainframe computer via a Web server on a Web server computer. The Web control program generates HTML through a completely table driven process, independent of file definitions and page definitions. The method further includes the steps of: having a user supply an identification code, forwarding the identification code from the Web server computer to the

mainframe computer, the Web control program then generating a state key, incorporating the state key into a Web page supplied to that user, and wherein the mainframe computer detects the state key from the Web page upon the user requesting data and decides whether that user can access the requested data based on the state key.

The method further includes the steps of: providing a database on the mainframe computer, the database having a database program that provides data in a given format; upon a user requesting data in the database, the Web server forwarding the request to the database, the database program accessing the data in the given format, and, by operation of the Web control program, generating a Web page with the requested data and supplying the generated Web page to the user.

Brief Description of the Drawings

The above and other features of the present invention will be more readily understood when the following detailed description is considered in conjunction with the accompanying drawings wherein like characters represent like parts throughout the several views and in which:

FIG. 1 is a simplified diagram of the arrangement used for the present invention;

FIG. 2 is a simplified view of the appearance of a Web page according to the present invention;

10

FIG. 3 is a simplified view of the appearance of an identification Web page according to the present invention;

FIG. 4 is a simplified view of the appearance of a restricted Web page according to the present invention;

FIGS. 5A and 5B are parts of a simplified overall flowchart of the handling of a Web request for data;

FIG. 6 is a flowchart of function determination, showing more details of parts of the flowchart of FIGS. 5A and 5B;

FIG. 7 is a flowchart of driven Web page data access; and

FIG. 8 is a simplified diagram illustrating a recursive Web page change feature of the present invention.

Detailed Description

Turning now to FIG. 1, some very basic concepts of the present invention will be discussed. Users accessing the World Wide Web of the Internet use their computers 10 running a Web browser program to access a Web server 12. (For ease of illustration, connections intermediate to the computers 10 and Web server 12, such as computers of an Internet service provider, are not shown.) The Web server 12 may be connected to receive information from another Web server such as 14. Additionally, the Web server 12 is connected to a mainframe computer 16 that

has legacy programs, legacy data, and legacy subroutines on it. (Legacy means existing information systems which may not be specifically adapted to Web access).

Previously, and as discussed in the background above, providing access to data on the mainframe via the Web required one to select from a number of unattractive options. The data could be duplicated on the Web server with the associated expansion in storage demands and increase in complexity in updating data (the data would have to be changed at each location where it is stored). Alternately, specific programs could be developed to allow Web access to particular data in the mainframe.

Advantageously, the present invention uses a Web control program that can generate Web pages in response to requests by users at computers 10. The Web control program runs on the mainframe 16 and generates the HTML needed to display particular data as part of a Web page. The Web access program on computer 16 additionally may, in response to a request for data that is not stored on mainframe computer 16, send a request for the data to another mainframe 18, receive the requested data back, put it into Web format (i.e., mark it up into HTML), and supply the Web page to the user. The mainframe computer 16 can access data at other computers by EDI (electronic data interchange), over TCP/IP (transport control protocol/internet protocol) or other protocols.

The Web control program is part of a Web management system also including a program that runs on the Web

12

server 12, which Web server is preferably running on a separate computer from the database-holding mainframe computer 16. In the discussion that follows, the Web management system will be referred to by the name EAGLE (Enhanced Application Generation Language for the Enterprise). EAGLE consists of several different and distinct parts including:

- There is a C program that runs as a CGI (common gateway interface) program on a Web server to extract and format data from forms that are submitted from a Web browser.
- There is a routing mechanism and a management structure for invoking CICS programs that can (Note: CICS is an IBM trademark.) produce Web pages. (As used herein, CICS stands for customer information control system running on the mainframe.) These pages can be created in any one of the three methods described below. Pages that are generated may then be interactively enhanced with styles, buttons, files to be read and/or updated, subroutines to be called, and many other variables using CICS-created objects.
- There is a state engine that keeps track of Web users and is able to maintain a session for Web users as well as do timeouts and re-establish

sessions using CICS programs.

- There is a CICS-based HTML generator that allows interactive creation of Web pages that can manipulate mainframe databases by reading, browsing, and updating VSAM and DB2. This one module can manage any number of different files, including fixed-length and segmented VSAM files, as well as DB2 databases. This program uses CICS-created objects for HTML generation including (but not limited to) file definitions, page definitions, drop-down lists, and Web page text.
- There is a set of development tools that facilitate the writing of EAGLE subroutines by generating object modules from simple templates that at the same time allow very flexible customizing of the modules. These object modules have the ability to read, browse, and update mainframe databases including VSAM (a specific brand name of indexed file structure) and DB2. (a specific database).
- Programs can be written and executed in any CICS-supported language without using any of the EAGLE development tools.
- This product can send and receive TCP/IP requests from any TCP/IP enabled source including Internet e-mail.

- Except for the C program that resides on the Web server, all of the programs are written in CICS Command Level Assembler language and run on IBM mainframes. This allows a level of security and scalability not found in other multi-tiered products. This would be especially useful in providing legacy mainframe systems and databases with Web access.
- All of the CICS objects necessary for operating EAGLE are created and maintained via the Web using EAGLE itself or by using mainframe terminal sessions (i.e., IBM3270).

Before supplying some specific examples of the Web management system in operation, some highly advantageous aspects of EAGLE will be discussed.

The combination of a CICS-based state engine and a program management structure that allows dynamic invocation of CICS programs means that the Web pages managed by EAGLE are associated with CICS-based objects that can be created and maintained interactively via the Web. Any Web page which is a part of EAGLE may call any subroutine, specify any files, pass variable data to the called subroutine, invoke additional subroutines or pages by way of return codes to the routing mechanism (error processing, rerouting of a request, pages by way of

return codes to the routing mechanism (error processing, rerouting of a request, etc.) These pages may require a PIN or other user authentication, invoke a process external to CICS, use a variable style-sheet, and include button objects. The flexibility of this system provides other features too numerous to mention here.

The ability to generate subroutines from CICS-based page templates that can be attached to EAGLE easily and seamlessly is another advantageous feature of EAGLE.

The existence of one single CICS program that can marry file definition objects and page definition objects resulting in the creation of a Web page that allows reading, updating, adding, deleting and browsing of VSAM files is an extraordinarily advantageous feature of the present invention. The EAGLE interactive HTML generator is completely table-driven and is independent of the actual file and page definitions.

The EAGLE subroutines may be written in any CICS supported language.

Presently, and in contrast to the present invention, the accessing of data on a mainframe from the Web is done using multi-tiered systems, most often, "screen-scraping". These systems routinely contain many different programming languages and require many different skill sets. These skill sets are often very different from the skill sets of the mainframe staff and require intense training and frequent use. Also, the multi-tiered systems require other computers to act as

proxies for the mainframe, which adds a significant overhead cost. EAGLE cuts through these issues by operating on the mainframe using the language skills already possessed by the mainframe staff. EAGLE runs on the mainframe so that no extra machines are required for accessing the mainframe from the Web. This reduces the maintenance costs dramatically and reduces the number of points of failure as well. This methodology also cuts out the use of extra software products that not only add overhead to any transaction but also increase the complexity of the application and increase the difficulty of debugging errors.

Turning now to FIG. 2, a specific example of a Web page that may be generated by the present invention is shown. In this and the following the Web pages will be discussed in the context of a university, but it will be readily understood that the invention has applicability to other contexts. FIG. 2 is a menu page where various menu items are hypertext, meaning that a computer user at a computer 10 (FIG. 1) can click on (put mouse cursor at and press enter or press left mouse button) and thereby select one of the menu items. For example, clicking on GRADES, will allow a university student to access his or her grades.

Before allowing access to the grades, the identification Web page of FIG. 3 requires the student to enter a student identification number and personal identification number. Upon the student supplying this

information, the Web server 12 (FIG. 1) passes it to mainframe 16. The mainframe 16 generates a state key and incorporates the state key into the next Web page (FIG. 4) displayed for the student. Specifically, the state key is randomly generated as a session key for this particular student. The state key is hidden on the Web page, meaning that a browser will not display it. However, when the student goes to another Web page restricted to that student (for example going from grades to transcripts), the manner of insertion of the state key on the Web page ensures that the browser program running on the computer 10 transmits the state key to the Web server 12. The Web server 12 in turn supplies the state key to the mainframe 16. Upon mainframe 16 confirming that the state key corresponds to the student in question, the transcript of the student can be accessed. Importantly, a part of EAGLE on the mainframe 16 disables a given state key if the state key is not received for a given time interval (timeout period). This reduces the chances that someone can readily access the private records of a student who steps away from his or her computer without remembering to close the program.

Other menu items of FIG. 2 may allow access to unrestricted information such as the course schedule for a coming semester. Such unrestricted information may be available without going through identification procedures or may require one to log in with a guest identification.

Turning now to the flowchart of FIGS. 5A and 5B (the latter being a continuation of FIG. 5A), the processing of a Web request for data will be discussed in detail. At block 20, a user at a computer 10 (refer momentarily back to FIG. 1) requests data from mainframe 16. The user's Web browser sends the request. At block 22, the Web server 12 formats the request such that mainframe 16 can respond. The mainframe 16 is running a CICS listener program such as supplied by IBM for use on its mainframes. At block 24, this passes control to CICS on the mainframe 16. Next, block 26 determines if the state key has been established and verified. If the particular data request requires a security, a screen such as FIG. 3 will prompt the user to enter information causing access authentication, and the association of access authentication to a state key. Next block 28 reads a primary function table to determine the functions required to supply the requested data.

The primary function table supplies the following:
program to execute, if applicable,
whether to LINK, XCTL, or START appropriate
program,
files/tables called programs to be used,
page definition table to be used,
additional subroutines to call,
route code definitions,
additional data to be used with program,
set update flag (Y/N, meaning yes or no),

PIN/Password protected page (Y/N) (some actions such as dropping a course may require reentry of PIN to increase protection against someone taking over when an authorized user has left without closing the Web browser program), and

call an external data source, if needed, supplies record key to table with all necessary information.

Block 30 then runs any program needed to respond to the request. If no program is called, block 32 simply formats a Web page and sends it back to the user via the Web server 12, which Web server may simply relay, without change, the Web page or may make minor insertions before supplying the Web page to the user. Block 32 puts together all the parts of the outgoing Web page including the page style, buttons, or other objects; the generated HTML, state key, and cache information. If XCTL or START is required, control goes to block 34 which then closes the connection.

If a LINK is made, a program on the mainframe is run. Block 36 reads incoming data using page definition table. Further block 38 allows subroutines to determine whether to continue with this program at block 42 or at block 40 set a route code and return to the main routing routine. Data access uses one or more of the following methods: dynamic page definition and file definition tables, and fixed object module(s).

20

Block 44 updates files if appropriate and if data passes all edits. Block 46 formats data for display using one or a combination of the following methods: dynamic page definition and file definition tables, and fixed object module(s).

Block 48 reads route code. If none is set, control goes to block 50 such that HTML is sent to the Web server, which in turn sends a response to the Web browser on computer 10.

If the route code is set, block 52 interprets the route code, and via block 54, returns to main routing section at block 28.

With reference now to FIG. 6, more details of block 28 of FIG. 5A will be given. At block 56, a primary function table 58 is read.

(A primary function table is given as Appendix 1 attached; a primary Web page format is attached as Appendix 2; a Web page forms and styles is attached as Appendix 3; a data access, edits, and layout definition is attached as Appendix 4; and a Web page object is attached as Appendix 5. These give specific examples of particular features used in the preferred embodiment.)

Block 56 leads to block 60 which tests if the page being requested is a protected page (i.e., restricted access). If so, block 62 returns a function to block 56 to test if the user is authorized. If block 60 determines that the page is not protected, control goes to block 64 which determines if pre-function checks are

21

needed. If yes, block 64 leads to blocks 66 and 68 which perform the checks and determine if the checks were passed. If so, the data appears proper and control goes to the continue at 70. If not, an error is noted at block 72.

Turning now to FIG. 7, a flowchart for driven Web page data access starts with block 74 which determines whether the page will use an object module. If yes, control goes to block 76 that provides Web page data access defined by the object module and leads to continue block 78. If no, a table driven process of data access at block 80 uses a primary Web page format 82, Web page objects 84, Web page forms and styles 86 and data access, edits, and layout definitions 88. Components 82 to 84 may be as shown on appendices 2 to 5.

FIG. 8 is a simplified example of how the present invention allows one, with proper authorization, to change a Web page in a recursive manner. The Web page 92 is displayed at a user's computer connected to Web server 12 and mainframe computer 16. The user uses the web browser program on his or her computer to send a page change command to server 12 and mainframe 16. This page change command, causes the generation of a revised version of the original Web page, this being changed Web page 94. The revision may include different formating, style, data, etc., but will replace the old Web page 92. The page change command, as used herein, changes the appearance of the original Web page, meaning that it

causes the Web page to appear differently to all who access it by causing it to display data from the computer storing the Web page. As used herein, "changes the appearance" or "changing the appearance" shall not include simply appending user comments or designs to a Web page as is common for Web-based bulletin board or public comment boards. By allowing recursive control (controlling a given Web page from that Web page being on a person's browser, not on the person's Web page editor program) of Web pages, the management of a large Web site can be quite efficiently performed.

Attached as appendices A, B, C, D, E, F, G, and H are copyrighted programs which run on a mainframe computer in a specific implementation of the invention. Attached as appendix J is a copyrighted program in C called Nirvana and which runs on the web server computer in a specific implementation of the invention. Certain non-critical information such as ports that are used on the actual implementation have been changed or left out to reduce exposure to unauthorized persons overcoming security measures in a specific implementation of the technique.

Although specific constructions have been presented herein, it is to be understood that these are for illustrative purposes only. Various modifications and adaptations will be apparent to those of skill in the art. In view of possible modifications, it will be

23

appreciated that the scope of the present invention
should be determined by reference to the claims appended
hereto.

Appendix I --PRIMARY FUNCTION TABLE

NERCTST5	MIDAS MENU SYSTEM	MMS 10/19/98	14:01:16	\$BO3
SELECT		DI6V		
Topic	Description	Group.		
MI-XSCRIPT				
Topic	Topic			
1	13.	COMADATA:	Auto?:	XCTL
2	14	COMAMORE:	Auto?:	Link
3	15.	COMAMOR2:		To
4	16.	DEFAULT	Start:	GATA MSYS
5	17.	Not Auth:	Update?:	N
6	18.	Not Open:	Student:	Y
7	19.	Invalid :	Targ/Hdme:	
8	20.	Return :		Not Available
9	21.	Process :		Phase(s) T
10.	22.	Choice :	Check	Holds Y Owner RA
11.	23.	Offuse :	Check	PIN Y MI-PIN
12.	24.	Xctl to :	Control	
			Cost	00000
Subroutines	File 1 CRDBI2		Chain	
1	File 2	Always Go	To	OK
2	File 3	Entry Required:		ERR
3	File 4	Notfnd	C:Prog	
4	File 5			
			PRT	
4-@	1 Sess-2	128.227.75.2	4/2	

Appendix II -- PRIMARY WEB PAGE FORMAT

NERCTST5	MIDAS	TEKT INPUT	MTE 10/19/98	14:01:21	BO3
SELECT		TOP 001	TOTAL LINES	008	DI6W
Topic	Description	Control	Dark Entry?		
MI-XSCRIPT					
Title:	FUN Transcript		Owner:	RA	Text
style=isis					
<!-- MI-XSCRIPT -->					
<hr>					
form=MI-GRADES					
form=MI-MENU					
</TABLE>					
FORM=MI-DECIDE					
			PRT		
4-@	1 Sess-2	128.227.75.2	5/2		

Appendix III -- WEB PAGE FORMS & STYLES

NERCTST5	M.Lus mn-*5E-eTs	MOB 10/19/98	14:01:24	\$BO3
SELECT	TOP 001	TOTAL LINES	009	DI6W
Topic	Description	Control	Dark Entry?	
MI-XSCRIPT		FORM		
Title:	Owner: RA			
		Text		
<!--MI-XSCRIPT-->				
<FORM METHOD="post" ACTION="/cgi-bin/cgiwrap/mlucas/nrvana">				
<INPUT TYPE=hidden" NAME="MDASTRAN" VALUE="MI-XSCRIPT">				
STATEKEY				
<INPUT TYPE="IMAGE" ALIGN="MIDDLE"				

```

SRC="http://www.isis.ufl.edu/wfimages/button.gif"
NAME="Transcripts"
</form>
Transcripts

```

PRT

4-© 1 Sess-2 128.227.75.2 5/2"

Appendix IV -- DATA ACCESS, EDITS AND LAYOUT DEFINITIONS

NERCTST5		MIDAS MODULE GENERATOR		MID	10/19/98 14:32:21	\$BO3
SELECT				008	001	17F
Program	Description	APPID	DelFld	Lins:	10	Max Segs: 146
RAMI0V00	MIDAS TEXT	MTEX	LINE	Qlins:	200	HedLen: 0449
Upd/Lst:	U C/M (CMB)	Fixed?	Log?:	LinLen:	100	SC-Offset: 0447
Typ (BSF):	SC Typ	Used?	Inseg?:	Limit:	100	SegLen: 0079
Blank Segs? Align:			C	T	LV-Offset: 0000	
Seq FLD	U M Out Ln Off	S H S I	Key	NR##?	p Ln	GR L P B Off Header
005 INPT	Y Y	10 000	H 1		10	00 0 S 0001 MIDAS Text
010 DESC	Y Y	40	H		40	00 0 D 0011 Description
015 CTRL	Y Y	10	H		10	00 0 D 0051 Control
020 ---		Y	01	H		01 00 0 E
025 TITL	Y Y	30	H		30	00 0 S 0247 Title
030 ---		Y	01	H		01 00 0 E
035 LINE	Y Y	79	S		79	00 0 S 0000 =
040 ---		Y	01	S		01 00 0 E

PRT

4-© 1 Sess-2 128.227.75.2 4/2"

Appendix V -- WEB PAGE OBJECTS

NERCTST5		GATA TEXT		TXT	10/19/98 15:24:37	SBO3
SELECT				KN		
GATA TEXT KEY						
GEN-ADD-7						

TEXT

7. SECTION ##### is a DUPLICATE of SECTION @@@@.
 You must drop @@@@ before adding another section of this same course.
 CALL SUBROUTINE SCHED

PRT

4-© 1 Sess-2 128.227.75.2 4/2"

Appendix A

TSO FOREGROUND HARD
DSNAME RA-PATENT

(RAMIO000)

* RAMIO000 TITLE 'READ FROM CICS TCPIP AND WRITE RECORDS TO A TD QUEUE'
* ALL INFORMATION IN THIS DOCUMENT, INCLUDING THE TRANSPORT
* MECHANISM AND PROTOCOLS, ARE COPYRIGHT 1998 BY THE
* UNIVERSITY OF FLORIDA. ALL RIGHTS RESERVED.

SUBROUTINE RAMIO000

FUNCTION:

READ A TCPIP SOCKET STREAM ORIGINATED FROM A CLIENT
DETERMINE PROPER FUNCTION TO PERFORM
LINK OR XCTL TO APPROPRIATE PROGRAM TO PERFORM THAT FUNCTION
RESPOND TO CLIENT WITH THE RESULTS

COPY	RARAM1S
COPY	RAMICOMM
COPY	RARAMDAS
COPY	RARAMMEN
USING	RARAMMEN,R2
 COPY	 RARATCPB
USING	RARATCPB,R12
 RAQUELEN	 EQU 32000
	DSECT
RAQUEUE	DS CL(RAQUELEN)
	COPY REGISTER DEFINITION
	COPY RARAREGS

DYNAMIC STORAGE

DFHEISTG	
COPY RARASPAD	
MIDASSTE	DS CL8
PEEKFLAG	DS F
TEMPMVAR	DS F
TEMPVLN	DS F
OPTVAL	DS OD
OPTVONOF	DS F
OPTVLEN	DS F
OPTLEN	DS F
OPTNAME	DS F
TIMEOUT	DS OD
TIMEOUTS	DS F
TIMEOUTM	DS F
*SNDMASK	DS D
*SNDMASK	DS D
*SNDMASK	DS D
*RETMASK	DS D
*RETMASK	DS D
*RETMASK	DS D
RSNDMASK	DS F
WSNDMASK	DS F
ESNDMASK	DS F
RRETMASK	DS F
WRETMASK	DS F
ERETMASK	DS F
MAXSOC	DS F

FLAGS	DS	F	FLAGS FOR SOCKET CALLS
DS	OF		
NAME	DS	CL16	SOCKET ADDRESS
NFAMILY	DS	H	TCP/IP ADDRESSING FAMILY
NPORT	DS	H	PORT DESCRIPTION
NADDRESS	DS	F	IP ADDRESS
NDZERO	DS	XL8	RESERVED (MUST BE ZEROS)
CLNTHNOL	DS	H	SOCKET DESCRIPTOR OF CLIENT MACHINE
BUF	DS	CL10	BUFFER FOR SOCKET READ/WRITE
C14TLBUF	DS	CLI20	CONTROL BUFFER FOR SOCKET READ
PHASE	DS	C	
PASSXCTL	DS	CL8	
SAVETOPC	DS	CL10	
DONETOPC	DS	CL10	
IDENT	DS	OCL16	
INITMVSA	DS	CL8	
INITADSN	DS	CL8	
PROCESS	DS	CL10	
SAVEFICE	DS	CL6	
TCPSVI4	DS	F	
TOTLQLEN	DS	F	
GOTONE	DS	C	
FACTSREQ	DS	C	
FACTWAIT	DS	C	
SVNBYTES	DS	F	SAVE SIZE OF BUFFER FOR SOCKET READ
SAVER14R	DS	F	SAVE REG 14 FOR BAL
MAXITEMS	EQU	90	
ITEMLEN	EQU	6	
QUEADDRS	DS	CL	((MAXITEMS)*(ITEMLEN))
EJECT			
TITLE 'READ FROM CICS TCPIP AND WRITE RECORDS TO A TD QUEUE'			
PROGRAM RAMI0000 STARTS HERE			
RAMI0000	DFHEIENT CODEREG=(7,8)		
RAMI0000	AMODE 31		
RAMI0000	PMODE ANY		
EXEC CICS HANDLE CONDITION DSIDERR(ERROR) x			
ERROR(ERROR) TERMIDERR(ERROR) SYSIDERR(ERROR) x			
ISCVNREQ(ERROR) INVRIQ(ERROR) IOERR(ERROR) x			
DISABLED(ERROR) ILLOGIC(ERROR) LENGERR(ERROR) x			
NOTAUTH(ERROR) NOSTG(ERROR) x			
EXEC CICS HANDLE CONDITION NOTFND(NOTFND) NOTOPEN(NOTOPEN)			
EXEC CICS ASSIGN STARTCODE(CODE)			
CLC	CODE,=C'SD'		
BNE	RETURN		
MVC	CONSTANT.CL8'DFHEIST' STORAGE LABEL		
MVC	PGRMNAM,=CL8'RAM10000' PROGRAM NAME		
RECIEVE INPUT FROM CICS TCPIP LISTENER			
A000	EQU	*	
MVC	TCPINLEN.=AL2(TCPIN\$)		
EXEC	CICS RETRIEVE INTO(TCPINPUT) LENGTH(TCPINLEN) x		
NONHANDLE			

```

CLC  TCPINLEN,=AL2(TCPIN$)  COMPARE LENGTH
BNE  PROGERR          BAD COMMAREA

MVC  SAVENOPE,BLANKS

DOTEST MVC  DATAAREA, DATAAREA

GET MAIN STORAGE FOR COMMAREA AND RAMDAS FILE

SR   R14,R14          CLEAR REG
LA   R14,COMAHDRL     GET CO"AREA LENGTH
AH   R14,=H'500' CHANGE THIS ONE ONLY
STH  R14,COMLTH
SR   R14-R:L4          CLEAR REG
LA   R14-MDASL        GET LENGTH FOR TOP OF MDAS
AH   R14,COMLTH        ADD TO GET TOTAL NEEDED
XC   RETCODE,RETCODE   CLEAR RETURN CODE
XC   BUF,BUF           CLEAR AREA
LA   R6,BUF            ADDRESS OF AREA
BAL  R14,GETRESPA     GET MESSAGE

L    R14,RETCODE
BCTR R14,0
LTR  R14,RI4
BM   NODATA
EX   R14-BUFBLANK     COMPARE TO BLANKS
BH   ERROR

NODATA EQU  *
      BAL  R14,SENDSEND
      BAL  R14,READSKT

      CLI  FACTSREQ,C'Y'
      BE   TESTAREA
      CLC  DATAAREA(4),C'TEST'
      BNE  NOTTEST
TESTAREA EQU  *
      MVC  DATAAREA,DATAAREA

* CHECK TO SEE IF THIS IS A CALL FOR INFORMATION ONLY

NOTTEST EQU  *
STATEMGT EQU  *
      EXEC CICS LINK PROGRAM 'RAM 0300' ) COMMAREA(RARAMDAS)      x
      LENGTH(CMLENGTH)
      BE   CHKWAIT          YES, CHECK TO SEE CLIENT IS WAITING.
      CLI  COMAPROF,C'P'    IS THIS A RESPONSE TO A REQUEST?
      BE   FACTCLOS         YES, RESPOND & CLOSE

      CLC  COMAMESS(6),C'VERIFY'  BLANK
      BNE  READ2ND

      CLC  COMATOPL,COMATOPC
      BE   CHECKEM
NORESOUR MVC  COMATOPC(10),C'MI-NORESRC'
      B    READ2ND

      CLC  BUF(9),C'CLOSE'   CLOSE?    COMMENTED OUT

```

```

      BNE  SOCKERR      NO, ERROR ERROR, OK

      MVC  SOCFUNC,CLI6'CLOSE      SOCKET FUNCTION = CLOSE
      CALL EZASOKET,(SOCFUNC,CLNTHNDL,      X
      ERRNO,RETCODE),
      VL,MF(E,PARMLIST)      X

      CLI  FACTSREQ,C'Y'      IS THIS AN INITIAL OUTSIDE REQUEST?
      BE   READ2ND      YES, PROCESS REQUEST

      MVC  TEMPMVNL,COMAMVNL      SAVE WRITE LENGTH

      MVC  COMAQADD,COMAMVAR      SET WRITE FROM ADDRESS
      PQUELOOP EQU  *
      MVC  COMAQLEN,TEMPMVNL+2      SET WRITE LENGTH
      B     PQUELOOP      GO WRITE ANOTHER QUEUE

      ENDQLOOP EQU  *
      EXEC CICS DEQ RESOURCE(COMAPQNM) LENGTH(8)

      BE     MOVETARG      YES, MOVE TARGET
      MVC  SAVEFICE,COMAFICE      SAVE FICE IN COMMAREA
      CLC  COMAFIC1,BLANKS      IS THERE A FICE CODE?
      NOQUE MVC  COMAPQNM,COMAMQNM
      MVI  COMAPQNM,C'='
      EXEC CICS LINK PROGRAM(COMAPROG) COMMAREA(COMMAREA) X
      LENGTH(COMACOML)
      B     PROGLINK

      READMEN EQU
      MVC  DONETOPC,COTIATOPC
      EXEC CICS READ DATASET('RAMMEN') SET(RZ) LENGTH(HALFWORD) X
      RIDFLD(COMATOPC)
      MVC  COMAOFLI,MMENOF1
      MVC  COMAMLDS,MMENNAME
      MVC  COMAS003,MMENS003
      MVC  COMAS004,MMENS004
      MVC  COMAS005,MMENS005
      MVC  COMAS006,MMENS006
      MVC  COMAS007,MMENS007
      MVC  COMAS008,MMENS008

      BR   R4

      VERIFYEQU *
      CLI  MMENPASS,C'Y'
      BNE  CKHOLDS
      MVC  COMAVERI,MMENPKEY

      B     READ2ND

      MVC  COMAIBFN,BLANKS
      MVC  COMACOLR,BLANKS
      EXEC CICS LINK PROGRAM(COMAPROG.) COMIMAREA(COMMAREA) X
      LENGTH(COMACOML)

```

IF THIS IS PROCESSING A WAITING REQUEST, FACTWAIT=Y, FACTSREQ=Y

```

USE2  PACK  DOUBLE,COMAIBFN(2)
DECR  SP    DOUBLE,*P'1'
CVB   R1,DOUBLE
MH    RI,*H,'10'
AR    R14,R1
CLC   0(10,R14), BLANKS

SENDSEND EQU

ST   R14,SPADSV14
A010 EQU *
L    R14.,SPADSVI4
BR   R14
READSKT EQU

ST   R14,SPADSVI4

READ RECORD FROM TCPIP SOCKET
CLC  RETCOD,=F'0'      ANYMORE BYTES
BE   SOCKERR          TCP/IP ERROR NO, GO CLOSE SOCKET
CHECK DIFFERENCE BETWEEN WHAT WAS EXPECTED AND WHAT WAS RECEIVED
L    R14,NBYTES        LOAD EXPECTED LENGTH
S    R14,RETCODE        GET DIFFERENCE
LTR  R14,R14           TEST REGISTER
MOVENUM MVZ  NUMBCHECK(*-),WKLEN    CHECK TO SEE IF NUMERIC
CHKNUM  CLC  NUMCHECK(*-*),NUMZONE  CHECK TO SEE IF NUMERIC
A21    EQU  *
MVC   NBYTES,=A(L'WKLEN) LENGTH OF AREA
L    R14,NBYTES
BCTR  R14,0
LTR   R14,R14
PUTRQST MVI  FACTSREQ,C'Y        MARK AS REQUEST AND
MVC   SOCKFUNC,=CL16'RECV'    SOCKET FUNCTION = READ
MVC   PEEKFLAG,=F'0'        SET TO PEEK AT DATA
LH   R14,=H'10'
A025  EQU  *
STH   R14.RECCOL.., SAVE RECV COUNTER
CALL  EZASOKET,(SOCFUNC,CLNTHNDL PEEKFLAG,
X
BNP   SOCKERR          IF NOT POSITIVE, ERROR
ST    R1,NBYTES         NUMBER OF BYTES NEEDED
B     A025

WAIT FOR CLIENT TO RESPOND

DORECVL EQU *
XC   RSNDMASK,RSNDMASK      INIT FIELDS
XC   WSNDMASK,WSNDMASK      "
RRETMASK,WRETMASK,ERETMASK,
X
ERRNOPRETCODE),
X
VL,MF=(E,PARMLIST)
L    RI,RETCODE
C    RL,=F'0'      POSITIVE RETURN CODE
BL    SOCKERR      NO. MUST BE AN ERROR
LH    R14.RECCOUNT  RESTORE LOOP COUNTER
BCT   R14,A025      GO ASK FOR MORE
B    SOCKERR        LOOPED MAX TIMES, ERROR

```

```

B020    EQU  *
B      READTEXT
NOTOPEN  EQU  *
CLC  SAVENOPE,BLANKS
BH  PUTNOPE
MVC  CCMATOPC(10),=C'NOTOPEN '
B      READTEXT
PUTNOPE  MVC  COMATOPC,SAVENOPE
B      READTEXT

SOCKERR  EQU  *
MVC  COMATOPC(10),=C'SOCKERR '
B      READTEXT

NOTFNDT  EQU  *
B      ERROR

NOTFND   EQU  *
CLC  COMAREGN,BLANKS
BH  PUTNOTF
MVC  COMATOPC(10),=C'NOTFOUND '
B      READTEXT
PUTNOTF  MVC  COMATOPC,COMAREGN
B      READTEXT

PROGERR  EQU  *
MVC  COMATOPC(10),=C'PROGERR '
BE  ERROR2      NO, GO TO ERROR

READQ    EQU  *
CLI  COMAPROF,C'S' FACTS RESPONSE
BNE  NOPROC

EXEC CICS LINK PROGRAM('RAMIIS00') COMMAREA(COMMAREA)
      LENGTH(COMACOML)          x
      LA  R0,* MOVE NULLS TO AREA
      LA  R14,QUEADDRS
      SR  R1,R1
      L  15,=A(L'QUEADDRS)
      MVCL R14,R0

      XC  TOTLQLEN,TOTLQLEN
READQAGN  EQU  *
      EXEC CICS LINK PROGRAM('RARASOKY') COMMAREA(COMMAREA)
      x
      LENGTH(COMACOML)
      LH  R14,COMAQITE      LOAD CURRENT ITEM NUMBER
      MVC  SOCFUNC,=C 'SELECT' SOCKET FUNCTION  =SELECT|
      CALL EZASOKET,(SOCFUNC,MAXsoc,TIMEOUT,
      BE  GETRBCK NO, GO BACK
      CALL EZACICO5,(BUF,NBYTES),VL,MF=(E,PARMLIST) TRAN 2 EBCDIC
      GETRBCK EQU
      RETURN EQU
      EXEC CICS RETURN
      EJECT
      TITLE 'ABENDS'
      COPY RAMIGETP
      QUEUEERR EQU
      MVC ABCODE+2,=C'QU'      INITIAL TSQUE READ ERROR

```

```

MVC BUF,MSGZOOB      ERROR MESSAGE FOR CLIENT
MVC NBYTES,=A(L'MSGZOOB) LENGTH OF ERROR MESSAGE
B ZOO                 ABEND
PROGERR EQU
  mvc ABCODE+2,=C'PR'  PROGRAM LOGIC ERROR
  MVC BUF,MSGZOOI      ERROR MESSAGE FOR CLIENT
  MVC NBYTES,=A(L MSGZOOI) LENGTH OF ERROR MESSAGE
  B ZOO                 ABEND
SOCKERRB EQU
  MVC ABCODE+2,=C'SO    TCP/IP SOCKET ERROR
  MVC BUF,MSGZOOIS     ERROR MESSAGE FOR CLIENT
  MVC NBYTES,=A(LMSGZOOIS) LENGTH OF ERROR MESSAGE
  B ZOO                 ABEND
NOTFNDT EQU
  MVC ABCODE+2,=C'TX'   TCP/IP SOCKET ERROR
  MVC BUF,MSGZOOT      ERROR MESSAGE FOR CLIENT
  MVC NBYTES,=A(LMSGZOOT) LENGTH OF ERROR MESSAGE
  B ZOOO ABEND
NOTFND   EQU
  MVC ABCODE+2,=C'NF'   RECORD NOT FOUND
  MVC BUF,MSGZOON      ERROR MESSAGE FOR CLINT
  MVC NBYTES,=A(LMSGZOON) LENGTH OF ERROR MESSAGE
  B ZOOO ABEND
ZOOO EQU
  B ZO10
ST     R14,ABR14      SAVE LAST RETURN ADDRESS
USING DFHEIBLK,R14 SET EIB DSECT BASE
L      R14,DFHEIBP LOAD EIB ADDRESS
MVC ABRESP,EIBRESP    SAVE LAST RESPONSE CODE
DROP R14   DROP EIB DSECT BASE
MVC ABCODE (2),PGRMNAM+4  SET UP ABEND CODE
EXEC CICS DUMP
  DUMPCODE(ABCODE)
  PROGRAM STORAGE TASK TERMINAL
  SPACE 2
  SEND ERROR MESSAGE TO THE CLIENT MACHINE
Z010  EQU
  MVC SOCFUNC,=CL16'SEND'   SOCKET FUNCTION=SEND
  MVC FLAGS,=F'0'  CLEAR FLAGS VARIABLE
  CALL EZACIC04,(BUF,NBYTES),VL,MF=(E,PARMLIST) TRANS. TO ASCII
  SPACE
  CALL EZASOKET,(SOCFUNC,CLNTHNDL,FLAGS,
  NBYTES,BUF,ERRNO,RETCODE),
  VL,MF=(E,PARMLIST)
  SPACE 2
  B   EZACLOSE   GO CLOSE TCPIP SOCKET AND EXIT
  SPACE 2
  TITLE 'CONSTANTS AND LITERALS'
  DS OF
  PACKONE  DC   PL4'1'  PACKED DECIMAL ONE
  BLANK    DC   X'40'
  BLANKS   DC   CL120'
  ZEROS    DC   CL30'0'
  CKTR     TR   COMATOPC(*-*),UPCASE   TRANSLATE TO UPPER CASE
  CASE
  UPCASE   DC   X'000102030405060708090A0B0C0D0E0F'
  DC   X'101112131415161718191A1B1C1D1E1F'
  DC   X'202122232425262728292A2B2C2D2E2F'
  DC   X'303132333435363738393A3B3C3D3E3F'
  DC   X'404142434445464748494A4B4C4D4E4F'

```

```

DC  X'505152535455565758595A5B5C5D5E5F'
DC  X'606162636465666768696A6B6C6D6E6F'
DC  X'707172737475767778797A7B7C7D7E7F'
DC  X'80C1C2C3C4C5C6C7C8C9A88B8C8D8E8F'
DC  X'90D1D2D3D4D5D6D7D8D99A9B9C9D9E9F'
DC  X'A0A1E2E3E4E5E6E7E8E9AAABACADAEEAF'
DC  X'B0B1B2B3B4B5B6B7B8B9ABBBCBDBEBF'
DC  X'C0C1C2C3C4C5C6C7C8C9CACBCCCDCECF'
DC  X'D0D1D2D3D4D5D6D7D8D9DADBDCDDDEDF'
DC  X'E0E1E2E3E4E5E6E7E8E9EAEBECEDEEEF'
DC  X'F0F1F2F3F4F5F6F7F8F9FAFBFCFDFF'
DC  OH
DS
UQUEFX  DC  CL2'LD'TD QUEUE PREFIX
MAX_SECONDS  DC  F'0000000'
FIONBIO  DC  XL4'8004A77E'
SO_REUSEADDR  DC  F'00000004'
SO_KEEPALIVE  DC  F'00000008'
SO_BROADCAST  DC  F'00000032'
SO_LINGER  DC  F'00000128'
SO_SNDBUF  DC  F'00004097'
SO_ERROR  DC  F'00004103'
SO_TYPE  DC  F'00004104'
NUMZONE  DC  CL28'00000000000000000000000000000000'
MAXWRK  DC  F'32767'      MAXIMUM RECORD LENGTH
MAXRLEN  DC  H'32717'      MAXIMUM RECORD LENGTH
MAXTIMES  DC  H'10'        MAXIMUM SENDS
FILEMSG  DC  C'OKTOSEND'  MESSAGE TO START SENDING DATA
OKMSG  DC  C'GOTFILEOK'  SUCCESSFUL MESSAGE
RECOCKMSG  DC  CL9'GOTDATAOK'  SUCCESSFUL RECORD MESSAGE
WAITMSG  DC  C'WITING FOR RESOURCES ON MAINFRAME...'WAIT MESSAGE
MSGZ00A  DC  C'ERROR: COMMAREA LENGTH ERROR'
MSGZ00B  DC  C'ERROR: TSQUE READ ERROR'
MSGZ00C  DC  C'ERROR: CCTR TSQUE READ ERROR'
MSGZ00D  DC  C'ERROR: LDL115P (EBCDIC->ASCII) ERROR'
MSGZ00E  DC  C'ERROR: CICS ERROR'
MSGZ00F  DC  C'ERROR: ONILOG VSAM WRITE ERROR'
MSGZ00G  DC  C'ERROR: TSQUE WRITE ERROR'
MSGZ00H  DC  C'ERROR: TRANSIENT DATA QUE ENQ PROBLEM'
MSGZ00L  DC  C'ERROR: PROGRAM LOGI ERROR'
MSGZ00N  DC  C'ERROR: MMEN ENTRY NOT FOUND
MSGZ00T  DC  C'ERROR: YOUR SESSION HAS TIMED OUT'
MSGZ00O  DC  C'ERROR: MMEN FILE NOT OPEN'
MSGZ00P  DC  C'ERROR: BAD RECORD LENGTH'
MSGZ00S  DC  C'ERROR: TCP/IP SOCKET ERROR'
MSGEND  EQU
LTORG
END  RAMI0000

```

CLC RETCODE,=F"0' ANYMORE BYTES
BE SOCKERR TCP/IP ERROR NO. GO CLOSE SOCKET

CHECK DIFFERENCE BETWEEN WHAT WAS EXPECTED AND WHAT WAS REC'D

L R14-NBYTES LOAD EXPECTED LENGTH

R14,NBITS LOAD EXPECTED
S R14,BTCODE GET DIFFERENCE

LTTR R14,REGCODE GET DIFFERENCE
LTTR R14,R14 TEST REGISTER

MOVENUM MVZ NUMCHECK(---) WHEN CHECK TO SEE IF NUMERIC

CHRONOLOGY

```
MVC NBYTES.=A(L'WKLEN) LENGTH OF AREA
L R14,NBYTES
BCTR R14,0
LTR R14,R14

PUTRQST MVI FACTSREQ,C'Y'      MARK AS REQUEST AND

MVC SOCFUNC.=CL16 'RECV'      SOCKET FUNCTION = READ
MVC PEEKFLAG.=F'0'           SET TO PEEK AT DATA
```

APPENDIX B

TSO FOREGROUND HARDCOPY
 DSNAME=RA.PATENT (RAMI0100)
 DATA SET RAMI0100
 ALL INFORMATION IN THIS DOCUMENT, INCLUDING THE TRANSPORT MECHANISM AND
 PROTOCOLS, ARE COPYRIGHT 1998 BY THE UNIVERSITY OF FLORIDA, ALL RIGHTS
 RESERVED

SUBROUTIN RAMI0100

FUNCTION:

ROAD, COMPRESS AN INCOMMING DATABLOCK.
 ADD ADDITIONAL FORMATTING AS DEFINED BY MTEX RECORD
 WRITE NECESSARY TS QUEUES

1.	RAMI0100	FILE RECORD SECT	
	COPY	RAMICOMM	SUBROUTIN COMMAREA
	USING	COMMAREA,RS	
	COPY	RARAMMEN	SUBROUTINE COMMAREA
	USING	RARAMMEN,R9	
	COPY	RARAMSTY	SUBROUTINE COMMAREA
	COPY	RARAMTEX	SUBROUTINE COMMAREA
	USING	RARAMTEX,R7	
	USING	MTEXS,R2	
	MAXAREA	EQU 32767	
	DATAAREA	DSECT	
2.	WORKING STORAGE		
	COPY DFHAID	3270 AID CHARACTERS	
	SPACE		
	COPY RARAREGS	REGISTER EQUATES	
	EJECT		
	DFHEIST DSECT	COMMAREA DSECT	
	SCRATCH PAD AREA		
	DOUBLE	DS D	
	FWORD	DS F	
	SAVEQADD	DS F	
	SAVER12	DS F	
	DATAQADD	DS F	
	DATAQLEN	DS H	
	KEEPQADD	DS F	
	KEEPQLEN	DS H	
	SAVEQLEN	DS H	
	HALFWORD	DS H	
	SAVEMAX	DS H	
	QREWRT	DS C	
	READTEXT	DS C	
	NEEDPRE	DS C	
	GOTNIRV	DS C	
	BEENHERE	DS C	
	REGID	DS CL4	
	SPADSKEY	DS CL30	
	STYLTYPE	DS C	
	BALWRITE	DS C	
	AFTHEAD	DS C	
	EJECT		
	PROGRAM RAMI0100 STARTS HERE		
	RAMI0100	DFHEIENT	CODEREG=(3,8), DATAREG=(13)
	RAMI0100	AMODE 31	
	RAMI0100	RMODE ANY	
		SPACE	
	CLC	EIBCALEN,=H'0'	

```

BE      RETURN
L      R5,DFHEICAP      GET COMMAREA
SPACE
EXEC CICS HANDLE CONDITION DSIDERR(ERROR)
      ERROR (ERROR) TERMIDERR(ERROR) SYSIDERR(ERROR)
      ISCVREQ(ERROR) INVREQERRQ IOERR(ERROR)
      DISABLED(ERROR) ILLOGIC(ERROR) LENGERR(ERROR)
      NOTAUTH(ERROR)
EXEC CICS HANDLE CONDITION NOTFND(NOTFND) NOTOPEN(NOTOPEN)
DEBUGDEBUGDEBUGDEBUG
      CLC  COMATOPC(10),=C'XXXXXXXXXXXX'
      BNE  AFTDBUG1
      DEBUG1  B  AFTDBUG1
      AFTDBUG1  EQU
DEBUGDEBUGDEBUGDEBUG
FIRST  EQU
      CLI  COMAPROF,C'S'
      BE   AFTFIRST
      MVC  COMAQITE,=H'1'
      MVC  COMAQ2DO,=C'DELETE'      SET TO DELETE QUEUE
      MVC  COMAQKEY,COMAMQNM      SET TS QUEUE NAME
EXEC CICS LINK PROGRAM('RARASOKY') COMMAREA(COMMAREA)
      LENGTH(COMACOML)
AFTFIRST  EQU
      USING MTEX,R2
      CLC  MTEXLINE(5),=C'STYLE'
      BE   MTEXLOOP
      CLC  MTEXLINE(5),=C'STYLE'
      BE   MTEXLOOP
      CLC  MTEXLINE(4),=C'FORM'
      BE   MTEXLOOP
      CLC  MTEXLINE(5),=C'<BODY'
      //www.isis.ufl.edu/wfirmages/logo2.gif"></td><h3>
      MVC  95(50,R12),MTEXNAME
      MVC  145(27,R12),=C'</h3></td></tr></table><hr>'
      MVI  172(R12),X'25'
      MVC  173(05,R12),=C'<PRE>'
      MVI  178(R12),X'25'
      LA   R12,179(R12)
      MVI  NEEDPRE,C'Y'
      CLC  MTEXLINE(4),=C'FORM'
      BE   MTEXFORM
      CLC  MTEXLINE(10),=C'#####'
      BNE  NOTDATA
      CLI  COMAMI01,C'Y'
      BE   AFTDATA
      SR   R15,R15
      LH   R1,DATAQLEN
      LR   R14,R12
      LR   R15,R1
      MVCL  R14,R0
      AH   R12,DATAQLEN
      ST   R12,SAVER12
      MVC  COMAQADD,r QADD
      LA   R2,MTEXLEN(R2)
      BCT  R6,MTEXLOOP
      B    AFTERTEX
NOTDATA  EQU
      CLC  MTEXLINE(08),=C'MDASTRAN'
      BNE  NOTTRAN
      CLC  COMATOPC,BLANKS

```

	BNH	BCTMTEX
	MVC	0(79,R12),MDASTRAN
	MVC	0(79,R12),CACHLINE
	MVC	44(8,R12),COMAMQNM
	MVI	79(R12), X'25'
	B	BCTMTEX
NOTNIR	EQU	
	MVC	0(79,R12), MTEXLINE
	MVI	79(R12),X'25'
BCTMTEX	EQU	
	LA	R2,MTEXSLEN(R2)
STYLE	EQU	
	EXEC	CICS HANDLE CONDITION NOTFND(NOTSY)NOTOPEN(NOSTY) DISABLED(NOSTY)
	LA	R14,9
	EX	R14,CKTR
	B	TITOVER
TITBCTLA	R15,1(R15)	
	BCT	R1,TITLOOP
TITOVER	EQU	
NOSUBS	EQU	
	CLC	MSTYLINE(8),=C'MDASTRAN'
	BNE	CKNIRV
	CLC	COMATOPC,BLANKS
	BNH	BCTMSTY
	MVC	0(79,R12), MDASTRAN
	MVC	0(79,R12),CACHLINE
	MVC	44(8,R12),COMMAMQNM
	MVI	79(R12),X'25'
	B	BCTMSTY
NOTNIRV	EQU	
CKBUTT	CLC	MSTYLINE(7),=C'MBUTTON'
	BNE	NOTMBUTT
	MVC	0(41,R12),-C'gif"><input type="hidden" names ="MDASTRAN" '
	MVC	41(27,R12),=C' value="XX- "></form>'
	MVC	49(10,R12),SPADSKEY
	MVI	68(R12),X'15'
	LA	R12,80(R12)
	B	BCTMSTY
NOTMBUTT	EQU	
	SR	R14,R15
	STH	R14,COMAQLEN
AFTTEXT	BAL	R4,COMPRESS GO AND COMPRESS THE AREA
	EQU	
	L	R14,COMAQADD
	AH	R14,COMAQLEN
	MVC	COMAQADD,SAVEQADD
COMPRESS	EQU	THIS CODE REMOVES THE EXTRA BLANKS
	LH	R15,COMAQLEN
	LTR	R15,R15
	BZ	NOAREA
	BNP	ERROR
	L	R14,COMAQADD
COMPLOOP	EQU	
	CLI	0(R14),X'00'
	A	R0,=F'100
		TEST15
NOTFND	EQU	
NOTOPEN	CLI	BEENHERE,
ERRR	EQU	
		EQU

```

MVI  COMAERRF,C'Y'
CLI  COMACONT,C'Y'
BE   MOVEMI01
MVI  COMAMI01,C'
B    CLRCONT
MOVEMI01 MVI  COMAMI01,C'Y'
CLRCONT  MVI  COMACONT,C'
DEBUGDEBUGDEBUGDEBUG
CLC  COMATOPC(10),=C'XXXXXXXX'
BNE  AFTDBUG3
DEBUG3   B    AFTDBUG3
AFTDBUG3  EQU
DEBUGDEBUGDEBUGDEBUG
MVC  COMAQADD,KEEPQADD
MVC  COMAQLEN,KEEPQLEN
RETURN  EQU
EXEC CICS RETURN
CONSTANTS AND LITERALS
BLANKS  DC  CL120'
CKTR   TR  SPADSKEY(*-),UPCASE      TRANSLATE TO UPPER CASE
UPCASE  DC  X'000102030405060708090A0B0C0D0E0F'
DC  X'101112131415161718191A1B1C1D1E1F'
DC  X'202122232425262728292A2B2C2D2E2F'
DC  X'303132333435363738393A3B3C3D3E3F'
DC  X'404142434445464748494A4B4C4D4E4F'
DC  X'505152535455565758595A5B5C5D5E5F'
DC  X'606162636465666768696A6B6C6D6E6F'
DC  X'707172737475767778797A7B7C7D7E7F'
DC  X'80C1C2C3C4C5C6C7C8C98A8B8C8D8E8F'
DC  X'90D1D2D3D4D5D6D7D8D99A9B9C9D9E9F'
DC  X'A0A1E2E3E4E5E6E7E8E9AAABACADAEEF'
DC  X'B0B1B2B3B4B5B6B7B8B9BABBBCBDBEBE'
DC  X'C0C1C2C3C4C5C6C7C8C9CACBCCCDCECF'
DC  X'D0D1D2D3D4D5D6D7D8D9DADBDCCDDDEF'
DC  X'E0E1E2E3E4E5E6E7E8E9EAEBECEDEEEF'
DC  X'F0F1F2F3F4F5F6F7F8F9FAFBFCFDFEFF'
DS   OH
MDASTRAN DC  CL79'<INPUT TYPE="hidden" NAME="MDASTRAN" VALUE="#####X
#####">
STTELINE  DC  CL79'<INPUT TYPE="hidden" NAME="MDASSTTE" VALUE=""
CACHLINE  DC  CL79'<INPUT TYPE="hidden" NAME="MDASCACH" VALUE=""
NEXTLINE  DC  CL79'<INPUT TYPE="hidden" NAME="MDASNEXT" VALUE="Y">
PREVLINE  DC  CL79'<INPUT TYPE="hidden" NAME="MDASPREV" VALUE="Y">
LTORG
END    RAMI0100

```

APPENDIX C

TSO FOREGROUND HARDCOPY		
DSNAME=RA.PATENT	(RAMI1B00)	
PROGRAM RAMI1B00		00010000
ALL INFORMATION IN THIS DOCUMENT, INCLUDING THE TRANSPORT		
MECHANISM AND PROTOCOLS, ARE COPYRIGHT 1998 BY THE		
UNIVERSITY OF FLORIDA. ALL RIGHTS RESERVED.		
SUBROUTINE RAMI1B00		00011013
FUNCTION:		00011113
RETRIEVE SPECIFIED DATA FROM TCP/IP REQUEST		00011213
DO NECESSARY DATA FORMATTING		00011313
RETURN RESULTS TO CALLING PROGRAM		00011413
		00011513
		00011613
		00011713
		00011813
		00011913
		00012013
		00012113
		00012213
		00012313
		00012413
		00012513
		00012713
		00012813
		00012913
		00013013
SASS MENU INTERFACE		00020000
		00030000
		00040000
COPY RACL0001		00041000
COPY RARAREGS		00042000
		00043000
		00044000
COMMAREA STARTS HERE BASE REG IS R5		00045000
COPY RAMICOMM	COPY COMMAREA DSCET	00046000
USING COMMAREA,R5		00047000
		00048000
FILE DESECT STARTS HERE BASE REG IS R9		00049000
		00050000
QLINES EQU 100		00060000
QLINELEN EQU 140		00070000
TSL EQU 5		00080000
		00090000
USING COMMAREA,R5		00100000
WORKING STORAGE		00110000
DFHEISTG DSECT	COMMAREA DSECT	00120000
		00130000
SCRATCH PAD AREA		00140000
FINDFLDT DS CL256		00141000
DOUBLE3 DS D		00142000
	DS OD	00143000
	DS F	00144000
ANSWER DC F'0'		00144100
DIVIDEND DS OD		00144200
		00144300
		00144400
		00144500
		00144600

QUOTIENT	DS	F	00144700
REMAINS	DS	F	00144800
DIVISOR	DS	F	00144900
MESS	DS	CL79	00145000
THISSEG	DS	PL3	00146000
CHAR9	DS	CL9	00147000
FLDLEN	DS	F	00147100
SV01	DS	F	00147200
SV014	DS	F	00147300
SV015	DS	F	00147400
SAVE6	DS	F	00147500
DEC01	DS	F	00147600
DEC14	DS	F	00147700
DEC15	DS	F	00147800
SIGDIGDS	F		00147900
FOUNDIT	DS	C	00148000
GOTDEC	DC	C	00148100
GOTSIG	DS	C	00148200
COPY RARASPAD			
PROGRAM RAMI1B00 STARTS HERE			
RAMI1B00	DFHEIENT	CODEREG=(3,8,7),DATARAEG=(13)	00148600
RAMI1B00	AMODE 31		00148700
RAMI1B00	RMODE ANY		00148800
	CLC	EIBCALEN,=H'0'	00148900
	BE	RETURN	00149000
EXEC CICS HANDLE CONDITION ERROR(ERROR)			
	L	R5,DFHEICAP	00149300
			00149400
			00149500
GETDATA	EQU		00149700
	BE	BCTLLOOP	00149800
	CP	COMAUSEG,=P'0'	00450700
	BE	BCTLLOOP	00450800
	CLC	14(8,R6),=C'MDASSEGN'	00450900
	BNE	BCTLLOOP	00451000
			00452000
	PACK	THISSEG,22(3R6)	00453000
	CP	COMAUSEG,THISSEG	00454000
	BNE	NOTSEG1	00455000
			00456000
	MVI	FOUNDIT,C'Y'	00457000
	B	BCTLLOOP	00458000
NOTSEG1	MVI	FOUNDIT,C'N'	00459000
	B	BCTLLOOP	00460000
		NOT THERE, BUMP A BYTE	00470000
			00480000
SAVR6 ST	R6,SAVE6		00490000
	CLI	FOUNDIT,C'Y'	00500000
	BNE	BCTLLOOP	00510000
	CLI	COMADECP,C'0'	00520000
	BNH	QUEMATCH	00530000
		ANY DECIMAL PLACES?	00540000
		NO, SKIP	00550000
	ST	R1,DEC01	00560000
	ST	R14,DEC14	00570000
	ST	R15,DEC15	00580000
		SAVE REGS	00590000
			00600000
PACK	DOUBLE2,COMADECPPUT	DEC PLACES IN DOUBLE?	00610000

JCL W/H			
CLC	COMAOLEN,NUMZONE	LENGTH	00620000
BE	DECEND	OK?	00630000
MVZ	NUMCHECK(2),COMAOLEN		00640000
CLC	NUMCHECK(2),NUMZONE		00650003
BNE	DECEND	NO, DONE	00660000
PACK	DOUBLE,COMAOLEN	PUT LENGTH IN DOUBLE	00670003
			00680000
			00690000
CP	DOUBLE,DOUBLE2 DEC PLACES BIGGER THAN LENGTH		00700003
BH	LOK		00710000
MVI	COMASERR,C'Y'		00720000
B	SAYERROR	ERROR	00730000
LOK	ZAP	SIGDIG,DOUBLE GET NUMBER OF	00731000
	SP	SIGDIG,DOUBLE2 SIG NUMERIC DIGITS	00732000
			00733000
			00734000
			00735000
	CVB	R1,DOUBLE PUT DEC PL IN R1	00736000
	ZAP	FWORD,=P'0' ZERO COUNTER	00737000
	LR	R14,R1 GET DEC PL	00738000
	BCTR	R14,0 SUB FOR EX	00739000
	EX	R14,MOVECH16 MOVE ZONES	00740000
	ZAP	DOUBLE3,FWORD GET INPUT COUNTER	00740100
	CVB	R1,DOUBLE3	00744800
	BNP	FDOK	00744900
FDZON	CLI	0(R15),C'	00745000
	BE	FDBCT	00745100
	CLI	0(R15),C'0'	00745200
	BL	FDERR	00745300
FDBCT	LA	R15,1(R15,)	00745400
	BCT	R1,FDZON	00745500
	B	FDOK	00745600
FDERR	MVI	COMASERR,C'Y'	00745700
	B	SAYERROR	00745800
FDOK	CVB	R1,DOUBLE FIND	00745900
	AR	R14,R1 DECIMAL	00746000
	CVB	R1,DOUBLE2 POINT	00746100
	SR	R14,R1 AREA	00746200
	BCTR	R14,0	00746300
			00746400
			00746500
			00746600
CP	LA	R15,22(R6) POINT TO INPUT	00746701
	FWORD,SIGDIG	INPUT MORE THAN SIG NUMERIC?	00746800
	BNH	USEFW NO USE INPUT COUNTER	00746900
	ZAP	DOUBLE3,SIGDIG GET SIG DIGITS	00747000
	B	GOTSD GO ON	00747100
USEFW	ZAP	DOUBLE3,FWORD GET INPUT COUNTER	00747200
GOTSD	CVB	R1,DOUBLE3 POINT	00747300
	AR	R15,R1 TO LAST BYTE	00747400
	BCTR	R15,0 OF INPUT DATA	00747500
			00747600
	CVB	R1,DOUBLE GET PLEN	00747700
	CVB	R0,DOUBLE2 GET DEC PTS	00747800
	SR	R1,R0 POINT	00747900
	LR	R0,R1 TO	00748000
	BCTR	R1,0 NUMERIC PORTION	00748100
	EX	R1,BLDEC BLANK IT	00748200
	LR	R1,R0 RESTORE REG	00748300
			00748400

MVL	CLI	0(R15),C'0'	DIGIT?	00748500
	BL	BCTDEC	DON'T MOVE	00748600
BCTDEC	MVC	0(1,R14),0(R15)	MOVE IN SIG DIGIT	00748700
	BCTR	R14,0	BUMP	00748800
	BCTR	R15,0	BACK	00748900
	BCT	R1,MVL	KEEP GOIN'	00749000
				00749100
DECEND	EQU			00749200
	CVB	R1,DOUBLE	GET PLEN	00749300
	BCTR	R1,0	FIX FOR EX	00749400
	EX	R1,OCDEC	MAKE ALL NUMERIC	00749500
	EX	R1,MVCDEC	MOVE TO R6	00749600
	LA	R14,22(R6)	BUMP TO	00749701
	MVI	0(R2),C'E'		01280000
	B		ERROR	01290000
				01300000
SERROK	EX	R14,PUTDATAE	DO MOVE INTO FINDFLDT	01310000
LH	R14,HALFWORD	RESTORE LENGTH OF TARGET FILED		01320000
	BCTR	R14,0		01330000
	LTR	R14,R14	TEST REGISTER	01330100
	BM	GOBACK	IF NEGITIVE DON'T MOVE IT	01330200
				01330300
	CLI	COMAFTYP,C'P'		01330400
	BE	DOPACKED		01330500
	CLI	COMAFTYP,C'B'		01330600
	BE	DOBINARY		01330700
				01330800
	EX	R14,CLCDATA	CHECK IF DATA HAS CHANGED	01330900
	BE	GOBACK	NOCHANGE, GOBACK	01331000
	EX	R14,MOVEDATA	MOVE DATA INTO TARGET FIELD	01331100
	CLC	COMAFNDF(4),=C'MDAS'		01331200
	BE	GOBACK		01331300
	B	SAYENT		01331400
				01331500
DOPACKED	EQU			01331600
	CLI	COMASERR,C'Y'		01331700
	BE	DATADONE		01331800
	MVC	CHAR9,NUMZONE		01331900
	LA	R1,CHAR9		01332000
	LA	R1,9(R1)		01332100
	MVZ	NUMCHECK(2),COMAOLEN		01332200
	CLC	NUMCHECK(2),NUMZONE		01332300
	BNE	FDERR		01332400
	CLC	COMAOLEN,NUMZONE		01332500
	BNE	OLOK		01332600
	EX	R14,PACKDA		01334400
	B	DATADONE		01334500
				01334600
DOBINARY	EQU			01334700
	CLI	COMASERR,C'Y'		01334800
	BE	DATADONE		01334900
	MVC	CHAR9,NUMZONE		01335000
	LA	R1,CHAR9		01335100
	LA	R1,9(R1)		01335200
	MVZ	NUMCHECK(2),COMAOLEN		01335300
	CLC	NUMCHECK(2), NUMZONE		01335400
	BNE	FDERR		01335500
	CLC	COMAOLEN,NUMZONE		01335600
	BNE	OLOK2		01335700
	MVC	COMAOLEN,COMAPLEN		01335800

OLOK2	PACK	DOUBLE2,COMAOLEN	01335900	
	CVB	R2,DOUBLE2	01336000	
	SR	R1,R2	01336100	
	EX	R2,LOADNUMB	01336200	
	PACK	DOUBLEE,CHAR9	01336300	
	CVB	R1,DOUBLE	01336400	
	CLC	COMAPLEN,=C'02'	01336500	
	BNH	DOHALF	01336600	
DOFULL	ST	R1,0(R15)	01336700	
	B	DATADONE	01336800	
DOHALF	STH	R1,0(R15)	01336900	
	B	DATADONE	01337000	
			01337200	
DATADONE	EQU		01337300	
			01337400	
SAYENT	MVI	COMAENT,C'Y'	01337500	
	B	GOBACK	01339300	
BLNCHECK	CLC	0(*-,R6),BLANKS	01339400	
PUTDATAE	MVC	FINDFLDT(*-,0(R6)	01339500	
CLCDATA	CLC	0(*-,R15),FINDFLDT	01339600	
MOVEDATA	MVC	0(*-,R15),FINDFLDT	01339700	
LOADNUMB	MVC	0(*-,R1),0(R6)	01339800	
PACKDATA	MVC	0(*-,R15),0(R1)	01339900	
			01340000	
ERROR	EQU		01340100	
	MVI	COMAERRF,C'Y'	01340200	
	MVI	COMASERR,C'Y'	01340300	
RETURN	DS	0H	SFW001	01340400
	MVC	COMADECP,BLANKS		01340500
	EXEC	CICS RETURN,	SFW001	01340600
				01340700
		CONSTANTS		01340800
ASIS DC	DC	C'N' Y=NO UPPER/LOWER CASE TRANSLATION	01340900	
BLANKS	DC	CL256'	01341100	
HEXZERO	DC	30X'00'	01341200	
NUMZONE	DC	24C'0'	01341300	
HEXFF	DC	30X'FF'	01342000	
			01343000	
CKTR	TR	0(*-,R6),UPCASE TRANSLATE TO UPPER CASE	01344000	
UPCASE	DC	X'000102030405060708090A0B0C0D0E0F'	01345000	
	DC	X'101112131415161718191A1B1C1D1E1F'	01346000	
	DC	X'202122232425262728292A2B2C2D2E2F'	01347000	
	DC	X'303132333435363738393A3B3C3D3E3F'	01348000	
	DC	X'404142434445464748494A4B4C4D4E4F'	01349000	
	DC	X'505152535455565758595A5B5C5D5E5F'	01350000	
	DC	X'606162636465666768696A6B6C6D6E6F'	01360000	
	DC	X'707172737475767778797A7B7C7D7E7F'	01370000	
	DC	X'80C1C2C3C4C5C6C7C8C98A8B8C8D8E8F'	01380000	
	DC	X'90D1D2D3D4D5D6D7D8D99A9B9C9D9E9F'	01390000	
	DC	X'A0A1E2E3E4E5E6E7E8E9AAABACADAECF'	01400000	
	DC	X'B0B1B2B3B4B5B6B7B8B9BABBBBCBDBEBF'	01410000	
	DC	X'C0C1C2C3C4C5C6C7C8C9CACBCCCDCCECF'	01420000	
	DC	X'D0D1D2D3D4D5D6D7D8D9DADBDCCDDDEDF'	01430000	
	DC	X'E0E1E2E3E4E5E6E7E8E9EAEBECEDEEEF'	01440000	
	DC	X'F0F1F2F3F4F5F6F7F8F9AFBFCFDFFFFF'	01450000	
			01460000	
	DS	0H	01470000	
			01480000	
LTORG			01490000	
END			01500000	

```

BE  TIMOUT1      YES, CONTINUE
CLC COMAVUID,BLANKS  HAS THERE BEEN A VALID USER?
BNH GOTMDAS      NO, CONTINUE
CLC COMAEXSS(9),COMAVUID  NO, IS IT THE SAME ONE?
BE  TIMOUT1      YES, CONTINUE
B   STARTOVR     NO, START OVER

TIMOUT1  EQU
ZAP  DOUBLE2,MDASLTIM  CHECK TO SEE IF TIMED OUT
MVC  MDASLTIM,DOUBLE  UPDATE LAST ACCESS TIME
AP   DOUBLE2,TIMEOUT1  |
CLC  DOUBLE,DOUBLE2   |
BNH  GOTMDAS      IT'S OK
BH   PUTCKEY      YES,CONTINUE
CLC  COMAEXSS,BLANKS  WAS COMAKEY THERE ALREADY?
BNH  GOTCKEY      YES, CONTINUE
MVC  COMAKEY(9),COMAEXSS  NO, PUT IT IN COMMAREA
B   GOTCKEY      YES, CONTINUE
MVC  COMAKEY(9),SAVEXSS  NO, PUT IT IN COMMAREA

PUTCKEY  MVI  COMAERRF,C'
MVI  COMACONT,C'
MVI  COMAMI01,C'
MVI  COMAPROF,C'
MVI  COMAFTYP,C'
MVC  COMAEDTC,BLANKS
ZAP  COMAUSEG,=P'0'
MVI  COMAPROF,C'

UNPK  COMAMQNM,EIBTASKN  COPY TD QUEUE PREFIX
OI   COMAMQNM+7,C'0'
MVC  COMAPROF,SAVEPROF
MVC  COMAMVAR,SAVEMVAR
MVC  COMAMVLN,SAVEMVLN
MVC  MDASLTIM,DOUBLE
MVC  COMACOML,COMLTH

LH   R14,=H'11'
LA   R15,COMAFICT      PUT DATA HERE
MVC  COMAFNDF,=C'FACTFICT'  DATA IDENTIFIER
BAL  R4,GETDATA      GO GET IT

LH   R14,=H'11'
LA   R15,COMAFIC1      PUT DATA HERE
MCV  COMAFNDF,=C'FACTFICE'  DATA IDENTIFIER
BAL  R4,GETDATA      GO GET IT

LH   R14,=H'10'
LA   R15,SAVETOPC      PUT DATA HERE
MVC  COMAFNDF,=C'MDASTRAN'  DATA IDENTIFIER
BAL  R4,GETDATA      GO GET IT
CLC  SAVETOPC,BLANKS
BNH  DQERROR      YES, CONTINUE
MVC  COMATOPC,BLANKS
MVC  COMATOPC(10),SAVETOPC

NOTMDAS EQU
DEBUGDEBUGDEBUGDEBUG
CLC  COMATOPC(10),=C'MI-PINT2'
BNE  AFTDBUG1
DEBUG1  B   AFTDBUG1
AFTDBUG1 EQU

```

	DEBUG	DEBUG	DEBUG	DEBUG
TRYAGAIN	EQU			
	EXEC CICS UNLOCK DATASET('RAMDAS')			
	MVC COMAPROC,SAVEPROC			
	MVI COMAPROC,C'S' FACTS RESPONSE			
	B DEQ			
DQERROR	EQU			
	MVC MDASKEYY,BLANKS			
	MVC COMATOPC,BLANKS			
	B DEQ			
DEQ	EQU			
	EXEC CICS DEQ RESOURCE(MIDASSTE) LENGTH(8)			
	CLI FACTSREQ,C'Y' FACTS REQUEST?			
	BNE RETURN NO, RETURN			
	MVC COMAPROC,SAVEPROC			
	B RETURN			
NORESRC	EQU	NO RESOURCES AVAILABLE		
	MVC COMAMESS,BLANKS			
	MVC COMAMESS(12),=C'NO RESOURCES			
ERROR	EQU			
	MVC MDASKEYY,BLANKS			
RETURN	EQU			
	EXEC CICS DEQ RESOURCE(MIDASSTE) LENGTH(8) NOHANDLE			
RETURN2	EQU			
	EXEC CICS RETURN			
	COPY RAMIGETP			
BLANKS	DC	CL130'		
NUMZONE	DC	15C'0'		
	LTORG			
	END	RAMI0300		

APPENDIX E

ALL INFORMATION IN THIS DOCUMENT, INCLUDING THE TRANSPORT MECHANISM AND PROTOCOLS, ARE COPYRIGHT 1998 BY THE UNIVERSITY OF FLORIDA. ALL RIGHTS RESERVED			00010000 00011003 00012003 00013003 00014003 00015003 00016003 00017003 00018003 00019003 00019103 00019203 00019303 00019403 00019503 00019611 00019703 00019803 00019903 00020203 00020303 00020403 00100000 00110000 00120000 00130000 00140000 00150000 00160000 00161008 00170000 00180000 00190000 00120000 00220000 00230000 00240000 00241000 00242000 00243000 00244000 00245000 00246000 00247000 00248000 00249000 00250000 00260000 00270000 00280000 00290000 003000000 00310000 00320000 00330000 00331000 00332000 00333000
SUBROUTINE RAM2A000			
FUNCTION:			
Read a incoming data and determine what file to read			00019611
Read a VSAM data set			00019703
Determine if the transaction is update, if so, update record			00019803
Format Web Page for browser			00019903
PRINT NOGEN			00110000
COPY RARAREGS			00120000
COPY RAMICOMM			00130000
COPY RARAMTEX			00140000
COPY RARAGATX			00150000
COPY RARAMSTY			00160000
COPY RARAMIOB			00161008
COPY RARAOLEG			00170000
USING COMMAREA,R5			00180000
COPY RARAQUED			00190000
QUEUE DSECT			00200000
QLINES	EQU 45	THESE TWO LINES NEED TO BE IN THE JOB	00210000
QLINELEN	EQU 200		00220000
QUEAREA	DSECT		00230000
QOUT	DS OCL500		00240000
QUELINE	DS CL500		00241000
QUELLEN	EQU *-QUEAREA		00242000
USING QUEAREA,R12			00243000
RADSECTO	DSECT		00244000
INSERTDS			00245000
DYNAMIC STORAGE			00246000
DFHEISTG	DSECT		00247000
COPY DFHAID			00248000
COPY RARASPAD			00249000
INSERTTT			00250000
UPAD	DS C		00260000
SCTY	DS C		00270000
NEWTOPC	DS C		00280000

AUTOADD	DS	C	003540001
DEFLD	DS	CL4	00335000
ALBS	DS	CL1	00336000
TRECLEN	DS	CL2	00337000
KEYOFF	DS	CL4	00338000
HEDLEN	DS	CL2	00339000
SEGLEN	DS	CL2	00340000
SEGOFF	DS	CL2	00350000
LEVOFF	DS	CL2	00360000
SEGCOUNT	DS	PL2	00370000
NEWSEGS	DS	PL2	00380000
ADDROLEG	DS	CL4	00390000
SADDOLEG	DS	CL4	00400000
SCNTOLEG	DS	CL4	00410000
APPID	DS	CL4	00411000
QLINES	DS	CL3	00411100
QLIMIT	DS	CL3	00411200
QLINELEN	DS	CL3	00411300
MAXSEGS	DS	PL3	00411400
FIXED	DS	C	00411500
QINSEG	DS	CL1	00411600
KEYLEN	DS	CL3	004117000
DISPLAY	DS	C	00411800
COURSE	DS	CL8	00411900
SECTION	DS	CL4	00412000
RDONLY	DS	CL1	00412100
SAVE9	DS	F	00412200
SVE10	DS	F	00412300
SAVE2	DS	F	00412400
SVE4	DS	F	00412500
SAVE7	DS	F	00412600
SAVE12	DS	F	00412700
SAVE14	DS	F	00412800
SAVE 15	DS	F	00412900
RDCNT	DS	PL3	00413000
RDMAX	DS	PL3	00413100
TAGMAX	DS	PL3	00413200
TAGCNT	DS	PL3	00413300
FINDFLD	DS	CL8	00413400
FINDFLDT	DS	CL79	00413500
THISSEG	DS	PL2	00413600
TEMP03	DS	CL3	00413700
OUTBLNK	DS	CL1	00413800
			00413900
			00414000
PROGRAM RAMI2A00 STARTS HERE			
RAMI2A00	DFHEIENT	CODEREG=(3,7,8)	00415000
RAMI2A00	AMODE 31		00416000
RAMI2A00	RMODE ANY		00417000
			00418000
			00419000
			00420000
START	EQU		00430000
	CLD	EIBCALEN,=H'0'	00440000
	BE	RETURN	00450000
	L	R5,DFHEICAP	00460000
	USING	COMMAREA,R5	00470000
	MVI	COMACONT,C'	00480000
			00490000
			00500000
	EXEC	CICS HANDLE CONDITION DSIDERR(ERROR)	00520000
	UNPK	KEYLEN,DOUBLE2	00805600

OI	KEYLEN+2,C'0'	00805700
BAL	R4,CLEANKEY	00805800
INSERTI		
PACK	DOUBLE,QLINES	00805900
CVB	R15,DOUBLE	00806000
PACK	DOUBLE,QLINELEN	00807000
CVB	R14,DOUBLE	00808000
ST	R14,FWORD	00809000
SR	R14,R14	00810000
M	R14,FWORD	00820000
STH	R15,COMAQLEN	00821000
EXEC CICS GETMAIN LENGTH(COMAQLEN) INITIMG(X'40') SET (12R)		
ST	R12,COMAQADD	00822000
LR	R14,R12	00823000
AH	R14,COMAQLEN	00824000
PACK	DOUBLE,QLINELEN	00825000
CVB	R15,DOUBLE	00826000
STH	R15,HALFWORD	00827000
SH	R14,HALFWORD	00828000
SH	R14,HALFWORK	00829000
ST	R14,COMAQ	00830000
MVC	COMAQLLN,HALFWORD	00840000
PACK	DOUBLE,QLINES	00850000
ZAP	RDMAX,DOUBLE	00860000
ZAP	TAGMAX,DOUBLE	00870000
ZAP	TAGCNT,=P'3'	00871000
ZAP	RDCNT,=P'0'	00872000
MVI	COMAERRF,C'	00873000
MVI	COMAENT,C'	00874000
MVI	COMACMD,C'	00875000
CLI	COMASTUD,C'A'	00876000
BE	CK4ADD	00877000
CLC	COMATOPL,COMATOPC	00878000
BE	SAMETOPC	00879000
MVI	NEWTOPC,C'Y'	00879100
B	READ2ND	00879200
SAMETOPC	LH R14,=H'1'	00879300
	LA R15,COMACMD	00879400
	MVC COMAEDTC,BLANKS	00879500
	MVC COMAEDIO,BLANKS	00879600
	MVC COMAFNDF,=C'MDASNEXT'	00879700
	MVI COMADECP,C'	00879800
	BAL R4,GETDATA	00880100
	MVI COMAERRF,C'	00880200
	SR R15,R15	00880300
	IC R15,0(R14)	00880400
	LA R15,1(15)	00880500
	STC R15,0(R14)	00880600
	STC R15,COMALEVL	00880700
	ZAP COMAUSEG,=P'0'	00880800
		01320000
		01330000
		01340000
		01350000
		01360000
		01370000
		01390000

	L R10,ADDROLEG	01400000
	USING RARAOLEG,R10	01440000
		01450000
		01460000
	ZAP DOUBLE,OLEGSC	01470000
	CVB R2,DOUBLE	01480000
	BNP ERROR	01481000
		01482000
	LA R10,OLEGHLEN(R10)	01483000
	USING OLEGS,R10	01484000
		01485000
UPLOOP	ZAP DOUBLE2,=P'0'	01486000
	EQU	01487000
		01488000
	CLC P;EGFMA,.=C'---'	01489000
	BE UPBCT	01490000
		01500000
	CLI OLEGWHER,C'S'	01510000
	BE UPSEGS	01510100
		01510200
	MVC COMAEDTC,OLEGEDIT	01510300
	MVC COMAEDIO,OLEGEDIO	01510400
	MVC COMAFTYP,OLEGFTYP	01510500
	PACK DOUBLE,OLEGFOFF	01510600
	CVB R14,COMAFWD3	01510700
	STH R14,DOUBLE	01510800
	LR R15,R9	01510900
	AH R15,COMAFWD3	01511000
	PACK DOUBLE,OLEGLENG	01512000
	AP DOUBLE2,DOUBLE	01513000
	CVB R14,DOUBLE	01514000
	AR R14,R15	STORAGE VIOLATION PROTECTION 01514111
	C R14,COMAREND	01514211
	BH BADADDR	01514311
	CVB R14,DOUBLE	01514411
		01514511
	MVC OMAFND(4),APPID	01514611
	MVC COMAFND+4(4),OLEGFNAM	01515100
	MVC COMAPLEN,OLEGLENG	01515200
	MVC COMAKEYF,OLEGHKEY	01515300
	MVC COMAFTYP,OLEGFTYP	01515400
	MVC COMAOLEN,OLEGOLEN	01515500
	MVC COMADECP,OLEGDEC	01515600
	LH R15,RECLEN	01515700
	CLI FIXED,C'Y'	01525200
	BE NOFADD1	01525300
	AR R15,R14	01525400
NOFADD1	STH R15,TRECLEN	01525500
	ST R2,SAVE2	01525600
	EXEC CICS GETMAIN SET(R2) LENGTH(TRECLEN) INITIMG(X'40')	01525700
		01525800
	LR R0,R2	01525900
	L R14,BFILADDR	01526000
	LH R1,HALFWORD	01526100
	MVCL R0,R14	01526200
	LR R9,R2	01526300
	ST R9,BFILADDR	01526400
	ST R9,SAVE9	01526500
	AH R9,HALFWORD	01526600
	SH R9,SEGLEN	01526700
	L R2,SAVE2	01526800

NOINS19	MVC	COMAEDTC,OLEGEDIT	01526900
	MVC	COMAEDIO,OLEGEDIO	01527000
	MVC	COMAFTYP,OLEGFTYP	01527100
	PACK	DOUBLE,OLEGFOFF	01527200
	CVB	R14,DOUBLE	01527300
	STH	R14,COMAFWD3	01527400
	LR	R15,R9	01527500
	AH	R15,COMAFWD3	01527600
	ST	R15,COMAFWD3	01527700
	PACK	DOUBLE,OLEGLENG	01527800
	AP	DOUBLE2,DOUBLE	01527900
	CVB	R14,DOUBLE	01528100
			01528200
	AR	R14,R15	01528300
	C	STORAGE VIOLATION PROTECTION	01528400
	BH	R14,COMAREND	01528600
	CVB	BADADDR	01528711
			01528811
	MVC	R14,DOUBLE	01528911
	MVC	COMAFNDF(4),APPID	01529011
	MVC	COMAFNDF+4(4),OLEGFNAM	01529111
	MVC	COMAPLEN,OLEGLENG	01529211
	MVC	COMAKEYF,OLEGHKEY	01529300
	MVC	COMAFTYP,OLEGFTYP	01529400
	MVC	COMAOLEN,OLEGOLEN	01529500
	MVC	COMADECP,OLEGDECP	01529600
	BAL	R4,GETDATA	01529700
	CLI	ADDFLAG,C'Y'	01529800
	BE	NODELSEG	01529900
			01530000
			01530100
			01530200
	BCT	R4,SEGULOP	01530300
SEGSDONE	L	R9,SAVE9	01540000
UPLOVER	EQU		01540100
			01540200
			01540300
			01540400
	CLI	COMAENT,C'Y'	01540500
	BNE	NOENT19	01540600
	CLI	COMAERRF,C'Y'	01540700
	BE	NOENT19	01540800
			01540900
			01541000
	CLC	SEGOFF,HEXZEROS	01541100
	BE	R14,SEGOFF	01541200
	LH	R14,SEGOFF	01541300
	LR	R15,R9	01541400
	AR	R14,R15	01541500
			01541600
			01541700
	CLI	SCTY,C'B'	01541800
	BNE	NOTSZB2	01541900
	LH	R15,0(R14)	01542000
	CVD	R15.DOUBLE	01543000
	AP	DOUBLE,NEWSEGS	01544000
	BH	OVERMAX	01545000
	CVB	R15.DOUBLE	01546000
	STH	R15.0(2,R14)	01547000
			01548000

	B	NONS19	01541000
			01550000
			01560000
			01570000
NOTSZB2	CLI	SCTY,C'Z'	01580000
	BNE	NOTSZ2	01581000
	PACK	DOUBLE,0(2,R14)	01582000
	AP	DOUBLE,NEWSEGS	01583000
	CP	DOUBLE,MAXSEGS	01583100
	BH	OVERMAX	01583200
	UNPK	0(2,R14),DOUBLE	01583300
	B	NONS19	01583400
			01583500
NOTSZ2	AP	0(2,R14),NEWSEGS	01583600
	CP	0(2,R14,MAXSEGS	01583700
	BH	OVERMAX	01583800
NONS19	EQU		01583900
	CLI	UPAD,C'N'	01584000
	BE	NOENT19	01584100
			01584200
		EXEC CICS REWRITE DATASET(COMAFILE) FROM(RARSECTO)	01584300
		LENGTH(RECLEN)	01584400
	MVI	REWRITE,C'R'	01584500
LOADSCRN	EQN		01584600
	CLC	COMAS001,BLANKS	01964000
	BH	DOS001	01964107
			01964207
	CLI	COMAERRF,C'Y'	01964307
	BNE	CKALGO	01965000
	CLC	COMAALGN,BLANKS	01966000
	BNH	NOALGO	01967000
	MVC	COMAIBFN,COMAALGN	01967100
	B	DOXXRET	01967200
			01967300
CKALGO	CLI	NEWTOPC,C'Y'	01967400
	BE	NOALGO	01967500
	CLC	COMAALGO,BLANKS	01967600
	BNH	NOALGO	01967700
	MVC	COMAIBFN,COMAALGO	01967800
	B	COXXRET	01967900
NOALGO	EQU		01968000
			01969000
		INSERT LOCATION S001	01970000
			01980000
			01990000
DOS001	CLC	COMAS001,BLANKS	02000000
	BNH	NOS001	02010007
			02020000
	MVC	COMAMSGF,BLANKS	02030000
	ST	R12,COMAUP12	02040000
	MVC	SPADSKEY,COMAS001	02050000
			02050108
	CLC	COMAS001(4),-C'RAMI'	02050208
	BE	NOTOBJ	02051008
			02052008
		EXEC CICS HANDLE CONDITION NOTFND(NOTOBJ) NOTOPEN(NOTOBJ)	02053008
		DISABLED(NOTOBJ)	02054008
		EXEC CICS READ DATASET('RAMIOB') SET(R1) RIDFLD(COMAS001)	02055008
		LENGTH(HALFWORD)	02056008
		USING RARAMIOB,R1	02057008

MVC	SPADSKEY.MIOBLOCK	02058000
EXEC	CICSA HANDLE CONDITION PGMDERR(BADSUBR)	02058114
EXEC	CICS HANDLE CONDITION NOTFND(NOTFND) NOTOPEN(NOTOPEN)	02059008
	DISABLED(NOTOPEN)	02059108
		02059208
NOTOBJ	EXEC CICS HANDLE CONDITION PGMDERR(NOS001)	02060008
	EXEC CICS LINK PROGRAM(SPADSKEY) COMMAREA(COMMAREA)	02061008
		02070000
	LENGTH(COMAUPL)	02080000
L	R12,COMAUP	02090000
CLC	COMAMSGF,=C'OK'	02100000
BNE	ERROR	02100106
CLI	NEWTOPC,C'Y'	02100206
BE	NOALGOS	02100306
CLC	COMAALGO,BLANKS	02100406
MVI	52(R12),X'15'	02714000
BAL	R10,TAGS	02715000
NONXPR19	EQU	02716000
		02717000
INSERT1	BEFORE MOVE	02718000
AUADD	EQU	02719000
		02720000
	CLI COMASTUD,C'A'	02730000
BNE	NOAUAD3	02740000
CLI	COMACMD,C'A'	02750000
BNE	NOAUAD3	02760000
MVC	COMAIBFN,=C'2'	02770000
B	RETURN	02780000
NOAUAD3	EQU	02790000
		02800000
	L R10,ADDROLEG	02810000
	USING RARAOLEG,R10	02820000
		02830000
ZAP	DOUBLE,OLEGSC	02831000
CVB	R2,DOUBLE	02832000
BNP	ERROR	02832100
LA	R10,OLEGHLEN(R10)	02832200
USING	OLEGS,R10	02832300
ZAP	DOUBLE2,=P'0'	02832400
		02832500
		02832600
		02832700
OPLOOP	EQU	02832800
		02832900
		02833000
CLI	OLEGWHER,C'S'	02834000
BE	SPLOOP	02835000
ZAP	COMAUSEG,=P'0'	02836000
		02837000
CKTABS	CLI OLEGTABL,C'S'	02838000
BNE	CKTABR	02839000
MVC	QOUT(07),=C'<TABLE>'	02839100
ST	R10,SAVE10	02939200
BAL	R10,TAGS	02839300
L	R10,SAVE10	02839400
MVC	QOUT(36),=C'<TR VALIGN="TOP" ALIGN="LEFT"><TD>'	02839500
MVC	QOUT(36),=C'<TR VALIGN="TOP" ALIGN="LEFT">	02839600
ST	R10,SAVE10	02839700
BAL	R10,TAGS	02839800
L	R10,SAVE10	02839900
		02840000

CKTABR	B	TABOVER	02840100
	CLI	OLEGTABL,C'R'	02840200
	BNE	CKTABD	02840300
	MVC	QOUT(36),=C'</TR><TR VALIGN="TOP" ALIGN="LEFT">'	02840400
	BNP	NOGATX	02851500
	LA	R2,GATXHLEN(R2)	02851600
	USING	GATXS,R2	02851700
			02851800
GATXLOOP	MVC	QOUT(79),GATXCOMM	02851900
	MVI	QOUT+79,X'15'	02852000
	ST	R14,FWORD	02852100
	ST	R10,SAVE10	02852200
	BAL	R10,TAGS	02852300
	L	R10,SAVE10	02852400
	L	R14,FWORD	02852500
	LA	R2,GATXSLEN(R2)	02852600
	BCT	R14,GATXLOOP	02852700
		NOGATXQEXECCICSHANDLECONDITIONNOTFND(NOTFND)NOTOPEN(NOTOPEN)	02852800
		DISABLED(NOTOPEN)	02852900
	MVC	0(5,R12),=C'</TD>'	02853000
	MVI	5(R12),X'15'	02853100
	ST	R10,SAVE	02852000
	BAL	R10,TAGS	02853300
	L	R10,SAVE10	02853400
	B	TABOVERE	02853500
			02853600
			02853700
TABOVERE	EQU		02853800
OPBCT	LA	R10,OLEGSLEN(R10)	02854000
	BCT	R2,OPLOOP	02854100
			02854200
	B	CKERRF19	02854300
SPLOOP	EQU		02854400
	ZAP	THISSEG,=P'0'	02854500
	ZAP	COMAUSEG,=P'1'	02854600
	ST	R10,SADDOLEG	02854700
	ST	R2,SCNTOLEG	02854800
	LH	R14,SEGOFF	02854900
	LR	R15,R9	02855000
	AR	R14,R15	02855100
	CLI	SCTY,C'B'	02855200
	BNE	NOTSZB4	02855300
NOTSZB4	LH	R15,0(R14)	02855400
	CVD	R15,DOUBLE	02855500
	ZAP	DOUBLE2,DOUBLE	02855600
	B	SKSZ4	02855700
	CLI	SCTY,C'Z'	02855800
	BNE	NOTSZ4	02855900
	PACK	DOUBLE,0(2,R14)	02856000
	STH	R14,COMAFWD3	02856100
	LR	R6,R9	02856200
	AH	R6,COMAFWD3	02856300
	PACK	DOUBLE,OLEGLENG	02856400
	AP	DOUBLE2,DOUBLE	02856500

CVB	R14.DOUBLE	02865512
AR	R14,R6 STORAGE VIOLATION PROTECTION	02865613
C	R14,COMAREND	02865712
BH	BADADDR	02865812
CVB	R14.DOUBLE	02865912
		02866012
MVC	COMAFNDF(4),APPID	02866100
MVC	COMAFNDF+4(4),OLEGFNAM	02866200
MVC	COMAREAS,OLEGHEAD	02866300
MVC	COMAPLEN,OLEGLENG	02866400
MVC	COMAKEYF,OLEGHKEY	02866500
SETOB19	MVI OUTBLNK,C'Y'	02878200
	AP COMAUSEG,=P'1'	02878300
	B SEGLOOP	02878400
		02878500
CKERRF19	EQU	02878600
		02878700
INSERT1	MVC	02878800
INSERT1	AFTER MOVE	02878900
		02879000
		02879100
ERR2ND	CLI COMAERRE,C'Y'	02879200
	BNE NOERR2	02879300
	EQU	02879400
		02879500
	CLI REWRITE,C'E'	02879600
	BE DUPE19	02879700
		02879800
	MVC SPADSKEY,BLANKS	02879900
	MVC SPADSKEY(10),=C,'INPUTERROR'	02880000
EXEC CICS HANDLE CONDITION NOTFND(N2GATX)NOTOPEN(N2GATX)		02880100
	RIDFLD(SPADSKEY)	02880200
		02880300
	USING RARAGATX,R2	02880400
	ZAP DOUBLE,GAT	02880500
	CVB R14.DOUBLE	02880600
	BNP N2GATX	02880700
	LA R2,GATXHLEN(R2)	02880800
	USING GATXS,R2	02880900
GATXL002	MVC QOUT(79),GATXCOMM	02881000
	MVI QOUT+79,X'15'	02881100
	ST R14,FWORD	02881200
	ST R10,SAVE10	02881300
	BAL R10,TAGS	02881400
	L R10,SAVE10	02881500
	L R14,FWORD	02881600
	LA R2,GATXSLEN(R2)	02881700
	BCT R14,GATXL002	02881800
N2GATX EXEC CICSHANDLECONDITION NOTFND(NOTFND) NOTOPEN(NOTOPEN)		02882000
	DISABLED(NOTOPEN)	02882100
B	NOUP19	02882200
		02882300
DUPE19	MVC 0(30,R12),=C'RECORD EXISTS,NOT ADDED'	02882400
	MVI 30(R12),X'15'	02882500
	AH R12,COMMAQLLN	02882600
	B NOUP19	02882700
		02882800
		02882900
NOERR2	EQU	02883000

			02883100
			03374100
	ZAP	DOUBLE,OLEGSC	03374200
	CVB	R2.DOUBLE	03374300
	BNP	ERROR	03374400
			03374500
	LA	E10,OLEGHLEN(10)	03374600
	USING	OLEGSR10	03374700
			03374800
PKLOOP	ZAP	DOUBLE2,=P'0'	03374900
	CLI	OLEGHKEY,C'	03375000
	BCG	PKBCT	03375100
			03375200
	LA	R15,COMAKEY+000	03375300
	PACK	DOUBLE,OLEGFOFF	03375400
	CVB	R14,DOUBLE	03375500
	STH	R14,COMAFWD3	03375600
	LA	R6,BLANKS	03375700
	PACK	DOUBLE,OLEGLENG	03375800
	AP	DOUBLE2,DOUBLE	03375900
	CVB	R14,DOUBLE	03376000
			03376112
	MVC	COMAFNDF(4),APPID	03377000
	MVC	COMAFNDF+4(4),OLEGFNAM	03378000
	MVC	COMAREAS,OLEGHEAD	03379000
	MVC	COMAPLEN,OLEGLENG	03380000
	MVC	COMAKEYF,OLEGHKEY	03380100
	MVC	COMAEDIO,OLEGEDIO	03380200
	MVC	COMACOLR,OLEGB1KY	03380300
	MVC	COMADECP,OLEGDECP	03380400
	MVC	COMABLZL,OLEGBLZL	03380500
	MVI	COMARAIN,C'Y'	03380600
	BAL	R4,PUTDATA	03380700
PKBCT	LA	R10,OLEGSLEN(10)	03380800
	BCT	R2,PKLOOP	03380900
			03381000
			03381100
			03381200
			03381300
	MVC	0(10,R12),=C'</FORM><P>'	03381400
	MVI	10(R12),X'15'	03381500
	AH	R12,COMAQLLN.	03381600
			03381700
	MVC	QOUT(09),=C'</TR><TR>'	03381800
	ST	R10,SAVE10	03381900
	BAL	R10,TAGS	03382000
	L	R10,SAVE10	03382100
			03382200
	MVC	QOUT(13),=C'</TR></TABLE>'	03382300
	ST	R10,SAVE10	03382400
			03382000
	BAL	R10,TAGS	03870000
	B	NOTAG19	03880000
			03890000
			03900000
ERROR	EQU		03910000
	MVI	COMAERRF,C'Y'	03920000
EXEC CICS HANDLE CONDITION NOTFND(ERNOTF) NOTOPEN(ERNOTF)			03930000
		DISABLED(ERNOTF)	03940000
	MVC	SPADSKEY,BLANKS	03950000
			03960000

	MVC	SPADSKEY(5),=C'ERROR'	03970000
	CLC	COMAREGN,BLANKS	03980000
	BNH	LEAVEEF	03990000
	MVC	SPADSKEY(8),COMAREGN	04000000
	LEAVEEFEXEC CICS READ DATASET('RAMSTY')SET(R2) LENGTH(HALFWORD)		
	USING	RARAMSTY,R2	04010000
	ZAP	DOUBLE,MSTYSC	04020000
	CVB	R14,DOUBLE	04030000
	BNP	ERNOTF	04040000
	LA	R2,MSTYHLEN(R2)	04050000
CKNOTFER	CLC	0(5,R2),=C'<FORM'	04060000
	BNE	CKSTUDER	04070000
	MVC	0(79,R12),0(R2)	04071000
	MVI	79(R12),X'15'	04072000
	AH	R12,COMAQLLN	04073000
	B	NOERBCT	04074000
			04075000
CKSTUDER	CLI	COMASTUD,C'Y'	04076000
	BE	NOSTU192	04077000
	BE	ERNOTF	04077002
			04077102
	CLI	COMAUPDA,C'N'	04078000
	BE	ERMDASAD	04079000
	CLI	UPAD,C'N'	04079100
	BE	ERMDASAD	04079200
	B	NOEROUT	04079300
ERMDASAD	CLC	NOERBCT	04079400
	BE	NOERBCT	04079500
			04079600
NOROUT	MVC	0(79,R12),0(R2)	04079700
	AH	R12,COMAQLLN	04079800
NOERBCT	LA	R2,MSTYSLEN(R2)	04079900
	BCT	R14,CKNOTFER	04080000
	B	NOSTU192	04080100
			04080200
			04080302
			04080400
ERNOTF	EQU		04080500
	B	NOSTU192	04080602
RETURN	EQU		04080700
	CLI	COMASTUD,C'A'	04080800
	BNE	DOXXRET	04080900
	CLI	COMACMD,C'A'	04081000
	BNE	DOXXRET	04081100
	MVC	COMAIBFN,=C'1'	04081200
	B	DOXXRET	04081300
DOXXRET	EQU		04081400
	EXEC	CICS RETURN	04081500
			04081600
			04081700
	COPY	RAMIGETP	04081800
	COPY	RAMIPUTP	04081900
			04082000
			04082100
CLEANKEY	LA	R10,COMAKEY	04083000
	SR	R2,R2	04084000
	IC	R2,50	04085000
	EX	R2,CKTR	04086000
		CLEAR REGISTER	04087000
		GET LENGTH OF FIELD ENTERED	04088000
		TRANSLATE FIELD TO UPPER	04089000
	LA	R10,COMAKEY	04090000
	LA	R15,50	

FIXHZ19	CLI	0(R10),X'00'	04100000	
	BNE	BCTHZ19	04110000	
	MVI	0(R10),C' '	04120000	
BCTHZ1019	LA	R10,1(R10)	04130000	
	BCT	R15,FIXHZ	04140000	
	BR	R4	04150000	
			04160000	
PUTERROR	MVC	QUELINE(*-*),0(R14)	01470000	
	COPY	RA02ERRM	04180000	
			04190000	
			04200000	
			04210000	
	TITLE	'CONSTANTS AND LITERALS'	04220000	
	DS	OF	04230000	
PACKONE	DC	PL4'1'	04240000	
BLANKS	DC	X'40'	04250000	
NUMZONE	DC	9C'0'	04260000	
HEXZEROS	DC	XL9'00'	04270000	
CKTR	TR	0(*-,R10),UPCASE	TRANSLATE TO UPPER CASE	04280000
UPCASE	DC	X'000102030405060708090A0B0C0D0E0F'	04290000	
	DC	X'101112131415161718191A1B1C1D1E1F'	04300000	
	DC	X'202122232425262728292A2B2CD2E2F'	04310000	
	DC	X'303132333435363738393A3B3C3D3E3F'	04320000	
	DC	X'404142434445464748494A4B4C4D4E4F'	04330000	
	DC	X'505152535455565758595A5B5C5D5E5F'	04340000	
	DC	X'606162636465666768686A6B6C6D6E6F'	04350000	
	DC	X'707172737475767778797A7B7C7D7E7F'	04360000	
	DC	X'80C1C2C3C4C5C6C7C8C98A8B8C8D8E8F'	04370000	
	DC	X'90D1D2D3D4D5D6D7D8D99A9B9C9D9E9F'	04380000	
	DC	X'A0A1E2E3E4E5E6E7E8E9AAABACADAECF'	04390000	
	DC	X'B0B1B2B3B4B5B6B7B8B9BABBBC8DBEBF'	04400000	
	DC	X'C0C1C2C3C4C5C6C7C8C9CCACBCDCECF'	04410000	
	DC	X'D0D1D2D3D4D5D6D7D8D9DADBDCCDDDEF'	04420000	
	DC	X'E0E1E2E3E4E5E6E7E8E9EAEBECEDEEEF'	04430000	
	DC	X'F0F1F2F3F4F5F6F7F8F9FAFBFCFDFEFF'	04440000	
SPACE			04450000	
	DC	OH	04460000	
			04470000	
			04480000	
END	RAMI2A00		04490000	
			04500000	

APPENDIX F

TSO FOREGROUND HARDCOPY			00010003
DSNAME=RA.PATENT			00020003
(RAMI1T00)			00030003
PROGRAM RAMI1T00			00031037
SASS MENU INTERFACE			00032037
ALL INFORMATION IN THIS DOCUMENT, INCLUDING THE TRANSPORT			00033037
MECHANISM AND PROTOCOLS, ARE COPYRIGHT 1998 BY THE			00034037
UNIVERSITY OF FLORIDA. ALL RIGHTS RESERVED			00035037
COPY RACL0001			00040003
COPY RARAREGS			00041003
COMMAREA STARTS HERE BASE REG IS R5			00042003
COMMAREA STARTS HERE BASE REG IS R5			00043003
COPY RARAGATX			00044003
COPY RAMICOMM			00045003
USING COMMAREA,R5			00046003
COPY COMMAREA DSECT			00047003
COPY COMMAREA DSECT			00047103
FILE DSECT STARTS HERE BASE REG IS R9			00047203
FILE DSECT STARTS HERE BASE REG IS R9			00047303
QLINELEN	EQU	140	00047403
TSL	EQU	5	00047503
QOUT	DSECT		00047603
	DS	CL79	00047703
DFHEISTG	WORKING STORAGE		00047803
	DSECT	COMMAREA DSECT	00047903
	SCRATCH PAD AREA		00048003
FOUNDIT	DS	C	00049003
FILENAME	DS	CL8	00050003
TEMPFLD	DS	CL79	00060003
FINDELDT	DS	CL79	00070003
	DS	OD	00080003
	DS	F	00090003
ANSWER	D	F'0'	00100003
DIVIDEND	DS	OD	00110003
QUOTIENT	DS	F	00120003
REMAINS	DS	F	00121003
DIVISOR	DS	F	00122003
MESS	DS	CL79	00123003
THISSEG	DS	PL3	00124003
FLDLEN	DS	F	00125003
AREAEND	DS	F	00126003
AWORD	DS	F	00127003
AREA1	DS	F	00128003
AREA14	DS	F	00129003
SV01	DS	F	00130003
SV014	DS	F	00131003
SV015	DS	F	00132003
SAVE4	DS	F	00133003
SAVE6	DS	F	00134003
SAVE2	DS	F	00135003

SAVE12	DS	F	00143123
DD2	DS	F	00143223
DDO	DS	F	00143323
DD1	DS	F	00143423
DD9	DS	F	00143623
DD14	DS	F	00143723
DD15	DS	F	00143823
SI00	DS	F	00143923
SI01	DS	F	00144123
SI14	DS	F	00144223
SI15	DS	F	
MOVEADDR	DS	F	00145003
MOVELENG	DS	H	00146003
DDSOFF	DS	H	00146103
RDMAX	DS	PL3	00146203
RDCNT	DS	PL3	00146303
TAGMAX	DS	PL3	00146403
TAGCNT	DS	PL3	00146503
DECIN	DS	CL30	00146603
DECOUT	DS	CL30	00146703
COPY RARASPAD			
PROGRAM RAMI1T00 STARTS HERE			
CLC	COMAEDIO,BLANKS		00150803
BNH	EDTOK		00150903
ST	R0,SI00		
00151003			
ST	R1,SI01		
00152003			
ST	R14,SI14		00153003
ST	R15,SI15		00154003
ST	R6,COMAUP01		00155003
ST	R12,COMAUP12		
00156003			
EXEC CICS LINK PROGRAM('RASI9200') COMMAREA(COMMAREA)			
LENGTH(COMACOML)			
L	R12,COMAUP12		00157003
00159003			
L	R0,SI00		00158003
00160003			
L	R1,SI01		
00170003			
L	R14,SI14		00180003
L	R15,SI15		00190003
B	NOSERRS		00200003
EDTOK EQU			
CLC COMABLZL,NUMZONE			
BNH	NOBLZL		00220003
LR	R1,R6		00230003
PACK	DOUBLE,COMABLZL		00240003
CVB	R0,DOUBLE		00250003
CLI	0(R1),C'0'		00260003
BNE	BCTBLZL		00261003
MVI	0(R1),C'		00261103
CKBLZL			
MVI	0(R1),C''		00261203
00261303			
CLC	GATXCOMM(8),=C'<SELECT'		00268523
BE	DOSEL		00268623
CLC	GATXCOMM(8),=C'<OPTION'		00268723
BNE	CKRADIO		00268823
PACK	DOUBLE,COMAPLEN		00268923
CVB	R15,DOUBLE		00269023
00269123			

	BCTR	R15,0	00269223
	EX	R15,CHEKOPTN	00269323
	BNE	NOTOPT	00269423
	MVC	TEMPFLD(17),=C'<OPTION SELECTED'	00269523
	MVC	TEMPFLD+17(62),GATXCOMM+8	00269623
	MVC	GATXCOMM,TEMPFLD	
		00269723	
	B	NOTOPT	00269823
			00269923
CKRADIO	CLC	GATXCOMM(19),=C'<INPUT TYPE="RADIO"'	00270023
	BNE	CKAREA	00270123
			00270223
	CLC	GATXCOMM+26(4),=C'@{@@@'	00270323
	BNE	NOTOAPI	00270423
	MVC	GATXCOMM+26(4),COMAAPID	
		00270523	
			00270823
NOTOAPI	PACK	DOUBLE,COMAPLENT	00270923
	CVB	R15,DOUBLE	00271023
	BCTR	R15,0	00271123
	EX	R15,CHEKRADI	
		00271223	
	BNE	NOTOPT	00271323
			00271423
	MVC	TEMPFLD(27),=C'<input type="RADIO" CHECKED'	00271523
	MVC	TEMPFLD+27(52),GATXCOMM+19	00271623
	MVC	GATXCOMM,TEMPFLD	
		00271723	
	B	NOTOPT	00271823
			00271923
CKAREA	CLC	GATXCOMM(16),=C'<textarea name="'	00272023
	BE	DOAREA	00272123
	CLC	GATXCOMM(12),=C'</textarea>'	00272233
	BNE	NOTOPT	00272323
	MVC	0(79,R12)GATXCOMM	00272423
	MVI	79(R12),X'15'	00272526
	BAL	R10,TAGS	00272626
	B	SKTOPT	00272730
			00272826
DOAREA	CLC	GATXCOMM+16(4),=C'@{@@@'	00272926
	BNE	NOARAPI	00273026
	MVC	GATXCOMM+16(4),COMAAPID	
		00273126	
			00273226
NOARAPI	MVC	0(79,R12)GATXCOMM	00273326
	MVI	79(R12),X'15'	00273432
	BAL	R10,TAGS	00273526
			00273626
	CLI	COMADECP,C'0'	00283126
	BNH	NODECP	00283226
			00283326
DOCHARP	MVC	DECIN,0(R6)	00283426
	BAL	R9,DECROUT	00283526
			00283726
	L	R15,FWORD	00283826
	AR	R12,R15	
		00283926	
	LA	R12,9(R12)	00284026
	LA	R14,1(R14)	00284126
	AR	R14,R0	00284226
	B	MOVEOVER	00284326

NODECP	EX	R15,MOVEDATO	00284426
	B	MOVEOVER	00284526
			00284626
			00284726
DOPACKED	EQU		00284826
	MVZ	NUMCHECK(2),COMAOLEN	00284926
	CLC	NUMCHECK(2),NUMZONE	00285026
	BNE	USEPLEN2	00285126
	CLC	COMAOLEN,NUMZONE	00285226
	BH	OLENOK	00285326
USEPLEN2	MVC	COMAOLEN,COMAPLEN	00285426
OLENOK	EQU		00285526
	PACK	DOUBLE,COMAPLEN	00285626
	CVB	R15,DOUBLE	00285726
	BCTR	R15,0	00285826
	B	SKIPUP	
		00285926	
UNPKPACK	UNPK	SPADSKEY(15),0(*-*),R6)	00286026
			00286134
SKIPUP	EX	R15,UNPKPACK	
00286226			
	OI	SPADSKEY+14,C'0'	00286326
	PACK	DOUBLE,COMAOLEN	00286434
	CVB	R15,DOUBLE	00286526
	BCTR	R15,0	00286626
	LA	R2,SPADSKEY	00286826
	LA	R2,15(R2)	00286926
	SR	R2,R15	00287034
	BCTR	R2,0	00287126
	CLI	COMADECP,C'0'	00287226
	BNH	NOPDEC	00287326
	LA	R2,1(R2)	00287426
	BCTR	R15,0	00287526
	EX	R15,UNPKDATP	00287626
	STH	R15,MOVELENG	00287726
	ST	R14,MOVEADDR	00287826
	MVC	DECIN,0(R2)	00287926
	CLI	COMAFTYP,C'B'	00288026
	BE	DIBINARY	00288126
	ST	R15,FWORD	00288226
	PACK	DOUBLE,COMAPLEN	00288326
	CVB	R15,DOUBLE	00288426
	BCTR	R15,0	00288526
	CLI	COMADECP,C'0'	00288626
	BNH	NODICP	00288726
	STH	R15,MOELENG	00288826
	ST	R12,MOVEADDR	00288926
	MVC	DECIN,0(R6)	00289026
	BAL	R9,DECROUT	00289126
	L	R15,FWORD	00289226
	AR	R12,R15	00289326
00297126			00289426
	LA	R12,9(R12)	00289526
	B	ENDTD	00289626

NODICP	EX R15.MOVEDITO AR R12,R15 00297626 LA R12,1(R12) MVI 0(R12),X'15' LA R12,1(R12) B NOSERRS	00297526 00297726 00297826 00297926 00298026 00298126 00298226
DECROUT	EQU MVC DECOUNT,7(R14) 00298326 PACK DOUBLE,COMADECP GET # OF DEC PLACES 00298426 CVB R0,DOUBLE PUT IN DOUBLE WORD EX R15,MODEDEC2 MOVE TO TEMP AREA SPAD24 00298626 SR R15,R0 GET LENGTH OF NON DEC PART EX R15,MODEDEC2 MOVE IT OUT LA R14,DECOUT LR R1,R14 LA R1,1(R1) AR R1,R15 STH R15,HALFWORD MVI 0(R1),C'.' LA R1,1(R1) LA R15,SPAD24 AH R15,HALFWORD LA R15,1(R15) 00299826 ST R14,AWORD LR R14,R0 EX R14,MODEDECO L R14,AWORD AR R1,R0 MVI 0(R1),X'15' LH R15,MOVELENG LA R15,1(R15) LA R15,2(R15) L R14,MOVEADDR EX R15,MODEMOVE DECROVER BR R9 MODEMOVE MVC 7(*-,R14),DECOUT MODEDEC2 MVC DECOUNT(*-),SPAD24 MODEDECO MVC 0(*-,R1),0(R15) 00301326	00298526 00298726 00298826 00298926 00299026 00299126 00299226 00299306 00299426 00299526 00299626 00299726 00299926 00300026 00300126 00300226 00300326 00300426 00300526 00300626 00300726 00300826 00300926 00301026 00301126 00301226
COLORS	CLI COMACOLR,C'.' BNH NOCOLR MVC QOUT(13),=C'<RONT COLOR=>' CKRED CLI COMACOLR,C'R' BNE CKBLUE MVC QOUT+13(05),=C'RED">' B COLRD CKBLUE CLI COMACOLR,C'B' BNE CKGREEN MVC QOUT+13(06),=C'BLUE">' B COLRD BCTR R15,0 UNPK SPADSKEY(9),DOUBLE2	00301426 00301526 00301626 00301726 00301826 00301926 00302026 00302126 00302226 00302326 00302426 00302526 00302626 00302726 00302826

OI	SPADSKEY+8,C'0'	00302926
LA	R2,SPADSKEY	00303026
LA	R2,9(R2)	00314926
SR	R2,R15	00315026
BCTR	R2,0	00315126
EX	R15,UNPKDITO	
	00315226	
AR	R12,R15	
	00315326	
LA	R12,1(R12)	00315626
MVI	0(R12),X	00315726
LA	R12,1(R12)	00315826
B	NOSERRS	00315026
		00316026
NOMOVER	LA R14,7(R14)	00316126
	B AMOVER	00316226
MOVER	AR R14,R15	
	00316326	
AMOVER	LA R14,,8(R14)	00316426
	MVC 0(2,R14),=C' "	00316526
	BAL R10,TAGC	00316626
		00316726
	MVC 00(25,R12),=C'SIZE="--" MAXLENGTH="-->"	00316826
	MVZ NUMCHECK(2),COMAOLEN	00316926
	CLC NUMCHECK(2),NUMZONE	00317026
	BNE USEPLEN	00317126
	CLC COMAOLEN,NUMZONE	00317226
USEPLEN	BH OLENOK2	00317326
	MVC 06(2,R12),COMAPLEN	00317426
	MVC 21(2,R12),COMAPLEN	00317526
	B LENDONE	00317626
OLENOK2	MVC 06(2,R12),COMAOLEN	00317726
	MVC 21(2,R12),COMAOLEN	00317826
LENDONE	MVI QOUT+25,X'15'	00317926
	CLC 06(2,R12),=C'01'	00318026
	BNE NOT01	00318126
	MVC 06(2,R12),=C'02'	00318226
NOT01	EQU	00318326
	MVI 25(R12),X'15'	00318426
	BAL R10,TAGC	00318526
		00318626
	MVC 0(5,R12),=C'</TD'	00318726
	BAL R10,TAGC	00318826
		00318926
	BAL R10,ERRROUT	00319026
		00319126
CLI	COMAERET,C'E'	00319226
BNE	NOSERRS	00319326
MVC	QOUT(38),=C'<td></td>'	00319426
MVC	0(09,R12),=C'</OPTION>'	00319526
MVI	9(R12),X'15'	00319626
L	R12,SAVE12	00319726
BAL	R10,TAGC	00319826
		00329726
LA	R14,SPADSKEY	
	00329826	
PACK	DOUBLE,19(4,R4)	00329926
CVB	R1,DOUBLE	00330026
AR	R14,R1	00330126
BCTR	R14,0	00330226
IC	R15,0(R14)	00330326

	LA	R15,1(R15)	00330426
	STC	R15,0(R14)	00330526
	B	DDREAD	00330626
			00330726
OPTOVER	EQU		00330826
ENODOPT	EQU		00330926
	L	R0,DD0	00331026
	L	R1,DD1	00331126
	L	R14,DD14	00331226
	L	R15,DD15	00331326
	BR	R2	00331426
			00331526
MOVEVAL	MVC	0(*-* ,R12),0(R9)	
00331626			
MOVEDISP	MVC	0(*-* ,R12),0(R9)	
00331726			
MOVEOKEY	MVC	SPADSKEY(*-*),0(R9)	00331826
CHEKDDS	CLC	0(*-* ,R14),R6)	00331926
			00332026
ERRROUT	EQU		00332126
	MVI	COMASERR,C'	00332226
	MVC	COMAERET,BLANKS	00332326
			00332426
			00332526
			00332626
			00332726
			00332826
			00332926
	L	R14,COMAUP14	00333026
	L	R15,COMAUP15	00333126
	L	R1,COMAUP	00333226
	ZAP	THISSEG,	00333326
			00333426
	L	STH R14,HALFWORD SAVE LENGTH OF TARGET FIELD	00333526
		R6,COMAMVAR POINT TO BEGINNING OF INCOMMING DATA	00333626
	L	R1,COMAMVLN GET TOTAL LENGTH OF THE DATA	00333726
	MVC	FINDFLDT,BLANKS	00333826
			00333926
QUELLOOP	EQU		00334026
			00334126
	CLC	COMAUSEG,=X'000000'	00334226
	BE	SAVR6F	00334326
	CP	COMAUSEG,=P'0'	00334426
	BE	SAVR6F	00335003
	B	CKFND	00340003
SAVR6F	MVI	FOUNDIT,C'Y'	00350003
			00360003
			00700003
			00710003
TAGC	EQU		00720003
TAGS	EQU		00721024
			00730003
NORES	C	R12,COMAQEND	00740003
	BNL	TAGENDC	00750003
	AP	TAGCNT,=P'1'	00760003
	AH	R12,COMAQLLN	00770003
	B	NOTAGC	00780003
			00790003
RESR12	LH	R14,COMAQLLN	00800003
	CLI	0(R12),X'25'	

00810003			
RESDONE	BE RESDONE	00820003	
	CLI 0(R12),X'15'	00830003	
	BE RESDONE	00840003	
	CLI 0(R12),C''	00850003	
	BH RESDONE	00860003	
	BCTR R12,0	00870003	
	BCT R14,RESR12	00880003	
	B NOTAGC	00890003	
RESDONE	EQU *	00900003	
	LA R12,1(R12)	00910003	
NOTAGC	BR R10	00920003	
TAGENDC	AH R12,COMAQLLN	00930003	
	MVI COMACONT,C'Y'	00940003	
	EXEC CICS LINK PROGRAM('RAMI0100')COMMAREA(COMMAREA)	00950003	
	LENGTH(COMACOML)	00960003	
	MVI COMACONT,C''	00970003	
	L R0,COMAQADD	00980003	POINT REG 0 TO SEGMENTS
	LH R1COMAQLEN	00990003	LOAD REG 1 WITH SEGMENT LENGTH
01020003		01000003	
	SR R15,R15	01010003	SET LENGTH TO ZERO
01030003			
	IC R15,=C''	01040003	SET PAD CHARACTER TO BLANK
	SLA R15,24	01050003	MOVE PAD CHARACTER TO BITS 1-8
	LA R14,*	01060003	R14 MUST BE A VALID ADDRESS
	MVCL R0,R14	01070003	MOVE PAD CHAR FOR LENGTH OF SEG
		01080003	
	CLI COMAQERR,C'Y'	01090003	SUCCESSFUL CREATE?
	BE ERROR	01100003	
01110003			
	L R12,COMAQADD	01120003	
	MVI COMACONT,C''	01130003	
	B NOTAGC	01140003	
		01150003	
		01160003	
		01170003	
ERROR	EQU	01180003	
01190003			
	MVI COMAERRF,C'Y'	01200003	
	MVI COMASERR,C'Y'	01210003	
	BR R10 GOBACK	01220003	
		01230003	
RETURN	EQU	SFW001	
01240003			
	MVC COMAOLEN,BLANKS	01241003	
	MVC COMAKEYE,BLANKS	01242003	
	MVI COMACOLR,C''	01243003	
	MVI COMADECP,C''	01244003	
	ZAP COMARDMX,RDMAX	01245003	
	ZAP COMARDCT,RDCNT	01245103	
	ZAP COMATGMX,TAGMAX	01245203	
	ZAP COMATGCT,TAGCNT	01245303	
	ST R12,COMAUP01	01245403	
		01245503	
		01245603	
	EXEC CIS RETURN	01245703	

		CONSTANTS	
ASIS	DC	C'N' Y=NO UPPER/LOWER CASE TRANSLATION	01245803
BLANKS	DC	CL133'	01245903
HEXZERO	DC	30X'00'	01246003
NUMZONE	DC	24C'0'	01247003
HEXFF	DC	30X'FF'	01248003
CKTR	TR	0(*-* ,R6),UPCASE	01249003
UPCASE	DC	TRANSLATE TO UPPER CASE	01250003
	DC	X'000102030405060708090A0B0C0D0E0F'	01251003
	DC	X'101112131415161718191A1B1C1D1E1F'	01252003
	DC	X'202122232425262728292A2B2C2D2E2F'	01253003
	DC	X'303132333435363738393A3B3C3D3E3F'	01254003
	DC	X'404142434445464748494A4B4C4D4E4F'	01255003
	DC	X'505152535455565758595A5B5C5D5E5F'	01256003
	DC	X'606162636465666768696A6B6C6D6E6F'	01257003
	DC	X'7071727377475767778797A7B7C7D7E7F'	01258003
	DC	X'80C1C2C3C4C5C6C7C8C98A8B8C8D8E8F'	01259003
	DC	X'90D1D2D3D4D5D6D7D8D99A9B9C9D9E9F'	01260003
	DC	X'A0A1E2E3E4E5E6E7E8E9AAABACADAEAF'	01261003
	DC	X'B0B1B2B3B4B5B6B7B8B9BABBBCBDBEBF'	01262003
	DC	X'C0C1C2C3C4C5C6C7C8C9CACBCDCECF'	01263003
	DC	X'D0D1D2D3D4D5D6D7D8D9DADBCDDDEDF'	01264003
	DC	X'E0E1E2E3E4E5E6E7E8E9EAEBECEDEEEF'	01265003
	DC	X'F0F1F2F3F4F5F6F7F8F9FAFBFCFDFF'	01266003
DS	OH		01267003
LTORG			01268003
END	RAMI1T00		01269003
			01410003
			01420003

APPENDIX G

ENTRY PONT IS: 11S00

ALL INFORMATION IN THIS DOCUMENT, INCLUDING THE TRANSPORT MECHANISM AND PROTOCOLS, ARE COPYRIGHT 1998 BY THE UNIVERSITY OF FLORIDA. ALL RIGHTS RESERVED.

COMMAREA DSECT
 EVENTUAL COMMAREA FOR THIS MODULE
 COPY RARAMDAS
 USING RARAMDAS,R9
 COPY RAMICOMM
 USING COMMAREA,R5
 COPY RARAMIDP
 COPY RARATCPB
 DEFINITION OF REGISTERS
 COPY RARAREGS
 CICS EXEC INTERFACE DYNAMIC STORAGE AREA DSECT

DFHEISTG
 COPY RARASPAD

SAVETRAN	DS	CL4	
NORESPND	DS	C	

WORKD DS D -- DOUBLE WORD CITATION NUMBER
 WORKD2 DS D -- DOUBLE WORD MESSAGE NUMBER
 WORKF DS F -- FULLWORD WORK AREA
 TEMPTLEN DS F -- FULLWORD WORK AREA

TEMPORARY STORAGE CONTROL

DSTSITM#	DS	H	B	ITEM NUMBER OF ENTRY IN QUEUE
DSTSLEN	DS	H	B	TEMPORARY STORAGE DATA AREA LENGTH
DSTSITPL	DS	H	B	ITEM NUMBER OF BIB 4 PALINK
DSTSDAAD	DS	A	ACON	TEMPORARY STORAGE DATA AREA ADDRESS
DSTSNAME	DS	CL8	AN	TEMPORARY STORAGE QUEUE NAME

PROGRAM CONTROL.

DSPCCALN	DS	H	B	COMMUNICATIONS AREA LENGTH
DSPCNAME	DS	CL8	AN	PROGRAM NAME
DSPCSAVE	DS	15F	AN	REGISTER SAVE AREA
SAVER14	DS	A		SAVE R14 ADDRESS
SAVER14A	DS	A		SAVE R14 ADDRESS
DSR14SAV	DS	A		SAVE R14 ADDRESS

STORAGE CONTROL

DSSCLEN	DS	H	B	STORAGE LENGTH
DSSCAD	DS	A	ACON	STORAGE ADDRESS
DSNXTPOS	DS	A		A(NEXT POSITION IN BUFFER)
DSSNDADD	DS	A		A(SEND BUFFER)
DSSNDEAD	DS	A		A(END OF BUFFER)
DSRECENT	DS	A		A(END OF RECORD)
DSSNDLEN	DS	H		LENGTH(SEND BUFFER)
DSDEST	DS	255C		DESTINATION
DSDSTLN	DS	H		LENGTH(DESTINATION)

NEW DATE STUFF

DSUTIME	DS	PL8		TIME FOR ASKTIME CALL
DSDOM	DS	F		DAY OF MONTH
DSDOW	DS	F		DAY OF WEEK
DSMOY	DS	F		MONTH OF YEAR
DSYEAR	DS	F		YEAR
DSDATE	DS	CL10		DATE
DSAPPL	DS	CL8		APPLID
DSTIME	DS	CL10		TIME
DSJC	DS	CL72		JOB STATEMENT

DSTRYCNT	DS	PL2	RETRY COUNT FOR HOST RESPONSE
DSCRFLAG	DS	C	CARRIAGE RETURN LINE FEED FLAG
FTEMP	DS	F	TEMPORARY FULL WORK
TCP-I STUFF			
DSNCOMM	DS	OF	COMM AREA FOR EZACIC25(GETHOSTBYNAME)
DSRETCOD	DS	F	RETURN CODE FROM GETHOSBYNAME CALL
DSERRNO	DS	F	ERROR NUMER
HOSTENT	DS	A	ADDRESS OF HOSTENT STRUCTURE
DSNCMND	DS	CL4	REQUESTED OPERATION (GHBN)
DSNAMLEN	DS	F	LENGTH OF NAME TO LOOK UP
DSNQTYPE	DS	CL1	TYPE OF QUERY
CSHFIRST	EQU	0	USE CACHE FIRST THEN GETHOSTBYNAME
DSNONLY	EQU	1	DO GETHOSTBYNAME CALL ONLY
CSHONLY	EQU	2	ATTEMPT QUERY USING CACHE ONLY
DSHNAME	DS	CL256	HOST NAME TO LOOK UP
DSNCOMM\$	EQU	*-DSNCOMM	
TCPINPUT			
SOCKDESC	DS	OF	INPUT FROM THE TCP/IP LISTENER
MVSADDR	DS	CL8	SOCKET DESCRIPTOR
TCPTASK	DS	CL8	MVS ADDRESS SPACE IDENTIFIER
DATAAREA	DS	CL35	TCP/IP TASK IDENTIFIER
FILLER	DS	C	CLIENT DATA AREA
SOCKADDR	DS	OF	FILLER
FAMILY	DS	H	SOCKET ADDRESS
PORT	DS	H	TCP/IP ADDRESSING FAMILY
ADDRESS	DS	F	PORT DESCRIPTION
DZERO	DS	XL8	IP ADDRESS
TCPIN\$	EQU	*-TCPINPUT	RESERVED (MUST BE ZEROS)
PARMLIST	DS	30A	
SOCFUNC	DS	CL16	SOCKET FUNCTION NAME
SOCRECV	DS	H	SOCKET DESCRIPTOR
PROTO	DS	F	SOCKET PROTOCOL
SOCTYPE	DS	F	SOCKET TYPE 1=STREAM,2=UDP
RETCODE	DS	F	RETURN CODE
ERRNO	DS	F	ERROR NUMBER
NBYTES	DS	F	SIZE OF BUFFER FOR SOCKET READ
FLAGS	DS	F	FLAGS FOR SOCKET CALLS
NAME	DS	OF	SOCKET ADDRESS
NFAMILY	DS	H	TCP/IP ADDRESSING FAMILY
NPORT	DS	H	PORT DESCRIPTION
NADDRESS	DS	F	IP ADDRESS
NDZERO	DS	XL8	RESERVED (MUST BE ZEROS)
CLNTHNDL	DS	H	SOCKET DESCRIPTOR OF CLIENT MACHINE
MAXSOC	DS	H	MAX. NUM OF SOCKETS OPEN AT ONE TIME
MAXNOS	DS	F	HIGHEST SOC NUM ASSIGNED TO APP.
AF	DS	F	ADDRESSING FAMILY (MUST = 2)
SUBTASK	DS	CL8	SUBTASK IDENTIFIER
BUF	DS	CL160	BUFFER FOR SOCKET WRITE
INBUF	DS	CL160	BUFFER FOR SOCKET READ
*SELECT CALL VARIABLES			
OPTVAL	DS	OD	
OPTVONOF	DS	F	
OPTVLEN	DS	F	
OPTNAME	DS	F	
TIMEOUT	DS	OD	
TIMEOUTS	DS	F	
TIMEOUTM	DS	F	
RSNDMSK	DS	F	READ SEND MASK

APPENDIX D

TSO FOREGROUND HARDCOPY (RAMI0300)
 DSNAME=RA.PATNET
 RAMI0300 TITLE 'STATE MANAGEMENT FOR MIDAS'

ALL INFORMATION IN THIS DOCUMENT, INCLUDING THE TRANSPORT MECHANISM AND
 PROTOCOLS, ARE COPYRIGHT 1998 BY THE UNIVERSITY OF FLORIDA. ALL RIGHTS RESERVED.
 SUBROUTINE RAMI0300

FUNCTION:

EVALUATE DATA FROM TCP/IP REQUEST

CREATE A NEW STATE KEY OR VALIDATE THE ONE THAT IS IN DATA

DSECTS

	DSECT
	COPY RAMICOMM
	COPY RARAMDAS
	COPY RARMIDP
	USING RARAMIDP,R12
MAXTRIES	EQU 10
	COPY REGISTER DEFINITION
	COPY RAREGS
DYNAMIC STORAGE	
	TEMP STORAGE HERE
	DFHEISTG
	COPY RARSPAD
COMLTH	DS H
CMLLENGTH	DS H
SAVETRY	DS H
NUMCHEK	DS CL15
UNPKTIME	DS CL15
MIDASSTE	DS CL8
UPDATE	DS C
TIMEOUT1	DS PL4
TIMEOUT2	DS PL4
TCPDSTAT	DS OCL16
TCPDSTTN	DS CL8
TCPDSSTTE	DS CL8
FINDFLD	DS CL8
FINDFLDT	DS CL79
UPDITEM	DS C
FACTSREQ	DS C
SAVETOPC	DS CL10
SAVEPROF	DS C
SAVEMVAR	DS CL4
SAVEMVLN	DS CL4
SAVEEXSS	DS CL10
SAVEPROC	DS CL10
STUID	DS CL20

PROGRAM RAMI0300 STARTS HERE

RAMI0300	DFHEIENT CODEREG=R,8)
RAMI0300	AMODE 31
RAMI0300	RMODE ANY

EXEC CICS HANDLE CONDITION DSIDERR(ERROR)

ERROR(ERROR)	TERMIDERR(ERROR)	SYSIDERR(ERROR)
ISINVREQ(ERROR)	INVREQ(ERROR)	IOERR(ERROR)
DISABLED(ERROR)	ILLOGIC(ERROR)	LENGERR(ERROR)

```

NOTAUTH(ERROR) DUPREC(NOMDAS)
EXEC CICS HANDLE CONDITION NOTFND(NOMDAS) NOTOPEN(ERROR)
START      EQU
CLC      EIBCALEN,=H'0'
BNH      RETURN2
L       R9,DFHEICAP
USING  RARAMDAS,R9
LA      R5,MDASLEN      GET LENGTH FOR TOP OF MDAS
AR      R5,R9           ADD IT TO R9
USING  COMMAREA,R5

MVC      COMLTH,COMACOML SAVE COMACOML
SR      R14,R14          CLEAR REG
LA      R14,MDASLEN      GET LENGTH FOR TOP OF MDAS
AH      R14,COMLTH        ADD TO GET TOTAL NEEDED
STH      R14,CMLNGTH     SAVE IT
SPACE
LH      R14,=H'10'
LA      R15,SAVEPROC      PUT DATA HERE
MVC      COMAFNDF,=C'PROCESS'  DATA IDENTIFIER
BAL      R4,GETDATA       GO GET IT
CLC      SAVEPROC,BLANKS
BNH      NOSTATE         YES, CONTINUE

EXEC CICS HANDLE CONDITION NOTFND(NOTPROC) NOTOPEN(NOTPROC)

LH      R14,=H'20'
LA      R15,STUID         PUT DATA HERE
MVC      COMAFNDF,=C'STUID'  DATA IDENTIFIER
BAL      R4,GETDATA       GO GET IT
MVI      COMAFTYP,C'U'
LA      R14,9            PUT DATA HERE
LA      R15,SAVEEXSS      PUT DATA HERE
MVC      COMAFNDF,=C'MDASSTID'
BAL      R4,GETDATA       GO GET IT

MVI      COMAFTYP,C'U'
LA      R14,9            PUT DATA HERE
LA      R15,SAVEEXSS      PUT DATA HERE
XC      NUMCHEK,NUMCHEK

!!!! HARD CODED TIMEOUT1 FOR NOW AT 5 MINUTES!!!!
CLI      TCPDSTTE,C'M'
BNE      NOMDAS
-----READ FILE FOR STATE CHECK-----
EXEC CICS READ DATASET('RAMDAS') SET(R2)
LENGTH(HALFWORD) RIDFLD(TCPDSTTE)UPDATE
CLC      CMLNGTH,HALFWORD   MAKE SURE LENGTH IS OK
BL       ERROR             ERROR IF TOO LONG FOR AREA
CLC      DOUBLE,DOUBLE2    I
BH       TIMEOUTL          IT HAS BEEN TOO LONG, START OVER
BNH      CKCONTN          NOT OVER, CONTINUE
EXEC CICS START TRANSID('MI06')
B       STARTOVR          IT HAS BEEN TOO LONG, START OVER
CKCONTN  EQU
CLC      SAVEXSS,BLANKS     WAS A NEW SSN ENTERED?
BNH      NONSSN            NO, CONTINUE
MVC      COMAEXSS(9),SAVEEXSS  YES, PUT IT IN COMMAREA
NONSSN   CLC      COMASIGN,=C   ADMINISTRATIVE USER?

```

WSNDMSK	DS	F	WRITE SEND MASK
ESNDMSK	DS	F	EXCEPTION SEND MASK
RRETMSK	DS	F	READ RETURN MASK
WRETMSK	DS	F	WRITE RETURN MASK
ERETMSK	DS	F	EXCEPTION RETURN MASK
SMAXSOC	DS	F	LARGEST SCK. DES., + 1
SR1	DS	F	SAVE R1
SAVEQLEN	DS	H	QUEUE LENGTH
SAVEQITE	DS	H	QUEUE ITEM NUMBER
CODE	DS	H	STARTCODE
COMMLEN	DS	H	INCOMMING COMMAREA LENGTH
FROMTCP	DS	C	STARTED FROM TCP/IP REQUEST
GOTPROC	DS	C	GOT A PROCESS READ
SAVER6	DS	F	SAVE REG 6
SAVER15	DS	F	SAVE REG 15
DATALENG	DS	F	LENGTH OF DATA FIELDS
GOTDATA	DS	C	LENGTH OF DATA FIELDS
PLACE	DS	PL1	LENGTH OF DATA FIELDS
WKLEN5	DS	CL5	CHARACTER FIELD FOR BUFFER LENGTH
WKLEN6	DS	CL6	CHARACTER FILED FOR BUFFER LENGTH
*			
EMAILTO1	DS	CL79	1 ST RECP EMAIL ADDRESS
EMAILTO2	DS	CL79	2 ND RECP EMAIL ADDRESS
EMAILTO3	DS	CL79	3 RD RECP EMAIL ADDRESS
EMAILTO4	DS	CL79	4 TH RECP EMAIL ADDRESS
EMAILTO5	DS	CL79	5 TH RECP EMAIL ADDRESS
EMAILFROM	DS	CL79	EMAIL ADDRESS OF SENDER
EMAILSUB	DS	CL79	EMAILSUBJECT LINE
MAINLINE	DS	OH	
	EXEC	CICS ASSIGN STARTCODE(CODE)	
	CLC	CODE,=C'SD'	
	BNE	STARTXCT	
	EXEC	CICS RETRIEVE SET(R5) LENGTH(COMMLEN)	
	MVC	EIBCCLEN,COMMLEN UPDAT EIBCALEN	
	B	GOTAREA	
STARTXCT	EQU		
	L	R5,DFHEICAP	LOAD ADDR COMM.AREA
	CLC	EIBCALEN,=H'0'	COMMAREA GREATER THAN 0
	BNH	RETURN	
GOTAREA	EQU		
	CLC	EIBALEN,COMACOML	
	BL	RETURN	
	UNPK	COMAMQNM,EIBTASKN	COPY TD QUEUE PREFIX
	OI	COMAMQNM+7,C'0'	
	BNH	RETURN	
	BAL	R14,IPAD2BIN	
CKRPORT	LA	R14,5	
	LA	R15SAVEPORT	PUT DATA HERE
	MVC	SAVEPORT,BLANKS	
	MVC	COMAFNDF,=C'PORTNUMB'	DATA IDENTIFIER
	BAL	R4,GETDATA	GO GET IT
	CLC	SAVEPORT,BLANKS	
	BNH	RETURN	
	PACK	DOUBLE,SAVEPORT	
	CVB	R14,DOUBLE	
	STH	R14,NPORT	

CKRINIT	EQU	•	
	XC	SAVEINIT,SAVEINIT	
	LA	R14,50	
	ST	R14,SAVELINI	INIT THE INIT STRING TO 50
	LA	R15,SAVEINIT	PUT DATA HERE
	MVC	COMAENDE,=C'RETNDATA'	DATA IDENTIFIER
	BAL	R4,GETDATA	GO GET IT
	CLC	SAVEINIT(50),BLANKS	WAS SOMETHING SENT?
	BL	RETURN	NOT AT LEAST BLANKS, GET OUT
	BH	GOTINIT	SOMETHING THERE, CONTINUE
	XC	SAVELINI,SAVELINI	NOTHING THERE, ZERO LENGTH
GOTINIT	EQU		
	BAL	R14,GETBUG	GETMAIN A BUFFER
	B	ATFGINFO	
NOTMIE	EQU		
	BAL	R14,GETBUF	GETMAIN A BUFFER
	BAL	R14,GETINFO	GET THE NECESSARY CALL INFO
AFTGINFO	EQU		
CONNECT TO COMMUNICATIONS SERVER			
	MVI	DSCRFLAG,C'	CLEAR<CR><LF> FLAG
	BAL	R14,CONNHOST	
	CLC	NPORT,=H'25'	
	BE	DOEMAIL	
	BAL	R14,SENDIHDR	
NOWAIT	BNE	CUTOFF	NO, BYE
CUTOFF	MVC	COMAIBFN,=C'3'	SEND REGULAR MESSAGE
	MVI	SAVEWAIT,C'N'	
	MVC	COMAPTR,COMATOPC	SAVE TARGET REQUEST
	LR	R0,R9	ADDRESS OF DATA
	AH	R0,=AL2(MDASLEN)	BUMP TO WHERE COMMAREA GOES
	LR	R14,R5	ADDRESS OF COMMAREA
	LH	R15,COMACOML	LENGTH OF COMMAREA
	LR	R1,R15	
	MVCL	R0,R14	MOVE THE DATA
	LH	R15,COMACOML	LENGTH OF COMMAREA
	AH	R15,=AL2(MDASLEN)	ADD LENGTH OF COMMAREA
	STH	R15,HALFWORD	SAVE LENGTH OF DATA
REWRMDAS	EQU		
	EXEC	CICS REWRITE DATASET('RAMDAS')	
		LENGTH(HALFWORD) FROM(RARAMDAS)	
	CLC	RETCODE,DFHRESP(NOTFND)	WAS IT NOT FOUND
	BE	GOCLOS	
	CLC	RETCODE,DFHRESP(NOTOPEN)	WAS NOT OPEN
	CLC	RETCODE,DFHRESP(DISABLED)	WAS IT DISABLED
	BE	GOCLOS	
	CLC	RETCODE,DFHRESP(NORMAL)	WAS IT NORMAL
	BNE	RETURN	
	B	GOCLOS	
PUTCONT	MVC	COMAMESS(8),=C'CONTINUE'	
	EXEC	CICS UNLOCK DATASET('RAMDAS')	

```

GOCLOS      EQU
             EXEC CICS DEQ RESOURCE (MIDASSTE) LENGTH(8) NOHANDLE
             BAL  R14.SENDCLOS
             B    RETURN

DOEMAIL      BAL  R14.SENDHELO
             BAL  R14,SNDMFROM
             BAL  R14.SNDMTO
             BAL  R14,SNDMDATA
             SR   R15,R15
             END TEST

IPAD2BIN     EQU
             ST   R14,SAVER14A
             LA   R14,SAVEIPAD
             LA   R15,NADDRESS
             ZAP  PLACE,=P'0'
             MADDLOOP   CLI  1(R14),C'
             BE   ZAP1
             CLI  1(R14),C'
             BE   ZAP1
             CLI  2(R14),C'
             BE   ZAP2
             CLI  3(R14),C'
             BE   ZAP3
             CLI  3(R14),C'
             BE   ZAP3
             B    NOADDR
             ZAP1      EQU  *
             PACK HALFWORD,0(1,R14)
             LA   R14,2(R14)
             B    ZAPADDR
             ZAP2      EQU  *
             PACK HALFWORD,0(2,R14)
             LA   R14,3(R14)
             B    ZAPADDR
             ZAP3      EQU  *
             PACK HALFWORD,0(3,R14)
             LA   R14,4(R14)
             B    ZAPADDR
             ZAPADDR   EQU  *
             ZAP  DOUBLE,HALFWORD
             CVB R1,DOUBLE
             STH R1,HALFWORD
             PUTADDIN   EQU  *
             MVC  0(1,R15),HALFWORD+1
             LA   R15,1(R15)
             AP   PLACE,=P'1'
             CP   PLACE,=P'4'
             BL   MADDLOOP
             NOADDR   EQU  *
             L    R14,SAVER14A
             BR   R14

```

ROUTINE TO GETMAIN AN AREA TO BE USED TO BUILD MAIL MESSAGE

SPACE 1

GOTMONTH	MVC	8(3,R2),4(R14)	MOVE IN THE MONTH TEXT
	MVC	12(4,R2),DSDATE+6	MOVE IN THE YEAR
	MVC	17(8,R2),DSTIME	MOVE IN THE TIME
		SPACE 1	
	AP	WORKD2,=PL8'1'	MESSAGE NUMBER FOR SUBJECT LINE
	L	R2,DSNXTPOS	START OF BUFFER
	LA	R2,SUBJ(R2)	WHERE TO MESSAGE NUMBER
	MVC	1(10,R2),DSDATE	DATE FOR SUBJECT LINE
	MVC	12(8,R2),DSTIME	TIME FOR SUBJECT LINE
	MVC	21(5,R2),=C'Part'	
	MVC	25(4,R2),=X'4021202020'	SET MASK
	ED	25(4,R2),WORKD2+6	CURRENT MESSAGE NUMBER
	L	R2,DSNXTPOS	START OF BUFFER
	LA	R2,DEST1@9R2)	WHERE TO WRITE DESTINATION
	LH	R15,DSDSTLN	LENGTH OF EMAIL ADDRESS
	LA	R4,DSDEST	A(DESTINATION)
	BCTR	R15,0	DECREMENT FOR EXECUTE
	EX	R15,MOVEDEST	MOVE EMAIL ADDRESS
	AH	R2,DSDSTLN	LENGTH OF EMAIL ADDRESS
	MVI	0(R2),C'>'	CLOSE RCPT TO: FIELD
L	R2,DSNXDEST	START OF BUFFER	
LA	R2,DEST2@(R2)	WHERE TO WRITE DESTINATION	
LH	R15,DSDSTLN	LENGTH OF EMAIL ADDRESS	
BCTR	R15,0	DECREMENT FOR EXECUTE	
EX	R15,MOVEDEST	MOVE EMAIL ADDRESS	
BR	R14		
CONNHOST	ROUTINE TO CONNECT TO SMTP.NERSP.NERDC PORT 25 VIA TCP/IP		
	EQU		
	ST	R1,SAVER14	
	MVC	SOCFUNC,=CL16'SOCKET'	SOCKET FUNCTION TO CALL
	MVC	AF,=F'2'	ADDRESSING FAMILY
	MVC	SOCTYPE,=F'1'	SET SOCKET TYPE TO STREAMS
	MVC	PROTO,=F'0'	USE DEFAULT PROTOCOL FOR STREAMS
	CALL	EZASOKET,(SOCFUNC,AF, SOCTYPE,PROTO,ERRNO,RETCODE), VL,MF=(E,PARMLIST)	
	CLC	RETCODE,=F'-1'	
	BE	LEOS	ERROR

TSO FOREGROUND HARDC'

	MVC	SOCRECV, RETCODE+2	SOCKET HANDLE FROM SOCKET CALL
	CLC	NADDRESS,=F'0'	CHECK TO SEE IF IP WAS SENT
	BH	GOTIPADR	YES, USE IT.
LOOKUP THE IP ADDRESS OF EMAIL HOST (GETHOSTBYNAME)			
	USING	RARATCPB,R12	USE TCP BUFFER DSECT TO ADDRESS
	MVZ	NUMCHECK96),COMATLEN	CHECK FOR NUMERIC IN TOTAL LENGTH
	CLC	NUMCHECK(6),NUMZONE	
	BNE	SOCKERR	
	PACK	DOUBLE,COMATLEN	PACK IT
	CVB	R14,DOUBLE	CONVERT TO BINARY
	A	R14,=A(TCPBLEN)	ADD LENGTH OF CONTROL BUFFER
	ST	R14,TEMPTLEN	SAVE IT
	CVD	R14,DOUBLE	CONVERT TO PACKED DEC
	UNPK	COMATLEN,DOUBLE	UNPACK IT
	OI	COMATLEN+5,C'0'	FIX LAST BYTE
	CLI	COMAPROF,C'Q'	Creating a request
	BE	BLDRQST	Yes, do it
	L	R12,COMACNTA	no, get the original cntl block
	MVC	TCPBBLEN,COMATLEN	SET TOTAL LENGTH TO BE SENT
	B	BLDRESP	
BLDRQST	LA	R12,CNTLBUF	SET ADDRESS OF AREA FOR SEND
	MVC	CNTLBUF,BLANKS	INIT TO BLANKS
	MVC	TCPBBLEN,COMATLEN	SET TOTAL LENGTH TO BE SENT
	MVC	TCPBTRAN,SAVETRAN	SET TRANSACTION TYPE
	MVC	TCPBSYST,=C'FED'	SET SYSTEM, FDET IMAG, ETC
	MVI	TCPBCOMM,C'0'	SET COMPRESS METHOD
	MVI	TCPBENC,C'0'	SET ENCRYPT METHOD
	MVC	TCPBREQS,=C'FLACENTSERV'	SET REQUESTING SERVER
	MVC	TCPBDATS,COMAFIC1	SET DATA HOST
	MVC	TCPBREQN,=C'01'	SET REQUEST TYPE NUMBER
	MVC	TCPBMESS,=C'01'	SET MESSAGE STATE
BLDRESP	EQU		
	BAL	R14,SENDHOST	SENT TO THE HOST
	L	R14,A=(TCPBLEN)	GET WHAT WAS SENT
	L	R15,TEMPTLEN	GET REMAINING LENGTH
	SR	R15,R14	SUBTRACT WHAT IS WAS SENT
	C	R15,=F'0'	IS THERE MORE?
	BE	NOMORED	NO, DO NOT READ QUEUE
	ST	R15,TEMPTLEN	SAVE WHAT IS LEFT
	L	R12,DSSNDADD	SET ADDRESS OF AREA FOR SEND
	ST	R12,COMAQADD	AREA TO BE FILLED BY READ
	MVC	COMAQ2DO,=C'READ'	SET TO READ QUEUE
	MVI	COMAQSTO	PUT IN ALLOCATED AREA
	MVC	COMAQKEY,COMAMQNM	SET TS QUEUE NAME
	MVC	COMAQITE,=H'1'	SET TO ITEM 1
	MVI	COMAQERR,C'N'	RESET QUEUE ERROR FLAG
GETQ4SND	EQU		
	MVC	COMAQLEN,=AL2(MAXSNDLN)	
	EXEC	CICS LINK PROGRAM('RARASOKY') COMMAREA(COMMAREA)	
		LENGTH(COMACOML)	
	CLI	COMAQERR,C'Y'	SUCCESSFUL READ?
	BE	SOCKERR	NO, GO TO ERROR
	LH	R14,COMAQLEN	SAVE AMOUNT READ
	ST	R14,NBYTES	SET TO SEND WHAT WAS READ
	L	R15,TEMPTLEN	GET REMAINING LENGTH
	SR	R15,R14	SUBTRACT WHAT IS BENIG SENT

	C	R15,=F'0'	IS THIS THE LAST ONE?
	BNE	NOTLAST	NO, DO NOT SET FLAG
NOTLAST	MVI	LASTSEND,C'Y'	SET LAST FLAG
	ST	R15,TEMPTLEN	SAVE WHAT IS LEFT
	BAL	R14,SENDHOST	SEND TO THE HOST
	LH	R14,COMAQITE	GET ITEM
	AH	R14,=H'1'	ADD ONE
	STH	R14,COMAQITE	PUT NEXT ITEM NUMBER
	CLI	LASTSEND,C'Y'	WAS IT THE LAST ONE
	BNE	GETQ4SND	NO, GET THE NEXT ONE
	BCT	R14,ENDLOOP	
MOVEEND	MVI	1(R15),C'>'	END BRACKET
	BR	R1	
BLANKR12	EQU		
	ST	R1,SR1	
	LA	R0,*	MOVE BLANKS TO AREA
	LR	R14,R12	
	SR	R1,R1	
	ICM	R1,8,=C'	
	LH	R15,=AL2(MAXSNDLN)	
	MVCL	R14,R0	
	L	R1,SR1	
	BR	R1	
NULLR12	EQU		
	ST	R1,SR1	MOVE BLANKS TO AREA
	LA	R0,*	
	LR	R14,R12	
	SR	R1,R1	
	ICM	R1,8,=X'00'	
	LH	R15,=AL2(MAXSNDLN)	
	MVCL	R14,R0	
	L	R1,SR1	
	BR	R1	
SNDMTO	EQU		
	ST	R14,SAVER14	
	CLC	EMAILTO1,BLANKS	
	BNH	SNDMTO2	
	BAL	R1NULLR12	
	MVC	NBYTES,=F'88'	LENGTH OF MESSAGE
	MVC	0(09,R12),TOMSG	FIRST PART OF MESSAGE
	MVC	09(79,R12),EMAILTO1	RECIPIENT
	BAL	R1,FINDEND	
	MVI	DSCRFLAG,C'Y'	SET <CR><LF> FLAG ON
	BAL	R14,SENDHOST	SENT TO THE HOST
	BAL	R14,HOSTRESP	WAIT FOR RESPONSE
SNDMTO2	EQU		
	CLC	EMAILTO2,BLANKS	
	BNH	SNDMTO3	
	BAL	R1,NULLR12	
	MVC	NBYTES,=F'88'	LENGTH OF MESSAGE
	MVC	0(09,R12),TOMSG	FIRST PART OF MESSAGE
	MVC	09(79,R12),EMAILTO2	RECIPIENT
	BAL	R1,FINDEND	
	MVI	DSCRFLAG,'Y'	SET <CR><LF> FLAG ON
	BAL	R14,SENDHOST	SENT TO THE HOST
	BAL	R14,HOSTRESP	WAIT FOR RESPONSE
		SEND THE MESSAGE HEADER AND GET THE RESPONSE FROM THE HOST	
SENDHDR	LA	R4,TOPHEAD#	NUMBER OF LINES IN THE HEADER
	MVC	NBYTES,=F'80'	LENGTH OF CARD IMAG
	BAL	R14,SENDHOST	SEND TO THE HOST
	S	R10,=F'80'	SUBTRACT FROM LENGTH

BAL	R14,HOSTRESP	WAIT FOR RESPONSE
LA	R3.80(R3)	NEXT CARD IMAGE
BCT	R4,SENDHDR	LOOP TILL END OF HEADER
SPACE 1		
SEND THE REST OF THE MESSAGE WITHOUT GETTING A RESPONSE FROM THE HOST		
SNDREST	EQU	
ST	R10,NBYTES	NUMBER OF BYTES TO SEND
MVC	SOCFUNC,=CL16'SEND	SOCKET FUNCTION = SEND
MVC	FLAGS,=F'0'	CLEAR FLAG VARIABLE
CALL	EZACIC04,((R3),NBYTES),VL,MF=(E,PARMLIST)	TRANS, TO ASCII
SPACE		
CALL	EZASOKET,(SOCFUNC,SOCRECV,FLAGS, NBYTES,(R3),ERRNO,RETCODE), VL,MF=(E,PARMLIST)	
SPACE		
L	R1,RETCODE	
CLC	R1,=F'-1'	POSITIVE RETURN CODE?
BE	LEOS	NO, MUST BE AN ERROR
BAL	R14,HOSTRESP	WAIT FOR RESPONSE
SPACE		
L	R14,SAVER14	RESTORE RETURN REGISTER ADDRESS
SEND QUIT COMMAND TO HOST		
SENDQUIT	EQU	
ST	R14,SAVER14	RESTORE RETURN REGISTER ADDRESS
MVC	0(1,R3),=C'QUIT'	MOVE IN QUIT COMMAND
MVC	NBYTES,=F'4'	LENGTH OF COMMAND
MVI	DSCRFLAG,C'Y'	SET <CR><LF> FLAG ON
BAL	R14,SENDHOST	SEND TO THE HOST
BAL	R14,HOSTRESP	WAIT FOR RESPONSE
L	R14,SAVER14	RESTORE RETURN REGISTER ADDRESS
BR	R14	RETURN
SEND CHUNK OF THE MESSAGE TO THE HOST (ADDING CRLF)		
SENDHOST	EQU	
ST	R14,DSR14SAV	SAVE RETURN REGISTER
MVC	SOCFUNC,=CL16'SEND'	SOCKET FUNCTION = SEND
MVC	FLAGS,=F'0'	CLEAR FLAG VARIABLE
MVC	BUF,0(R12)	SEE WHAT'S GOING
TRANSEND	EQU	
TRANSLATE TO ASCII WITH EZACIC04		
CALL	EZACIC04,((R12),NBYTES), VL,MF=(EPARMLIST)	TRAN TO ASCII
SPACE		
CLI	DSCRFLAG,C'Y'	<CR><LF> FLAG SET ?
BNE	EZASEND	NO, JUST SEND WHAT IS IN BUF
SPACE		
L	R14,NBYTES	NUMBER OF BYTES
AR	R14,R12	BUMP POINTER PAST TEXT
MVC	0(2,R14),=X'OD0A'	MOVE IN ASCII <CR><LF>
L	R14,NBYTES	NUMBER OF BYTES
SPACE		
CLC	RETCODE,NBYTES	DID SEND IT ALL
BNL	SENTALL	YES, CONTINUE
CLC	RETCODE,=F'-1'	CHECK RETURN CODE
BNH	LEOS	TCP/IP ERROR
L	R1,RETCODE	NUMBER OF BYTES SENT
AR	R12,R1	BUMP THE RECORD POINTER
L	R1,NBYTES	NUMBER OF BYTES INTENDED
S	R1,RETCODE	SUBTRACT THE NUMBER SENT

	LTR	R1.R1	TEST REGISTER
	BNP	R1.NBYTES	IF NOT POSITIVE, ERROR
	ST	R1.NBYTES	NUMBER OF BYTES STILL LEFT
	LH	R14.HALFWORD	CHECK LOOP COUNT
	BCTR	R14,0	DECREASE BY ONE
	LTR	R14.R14	TEST REGISTER
	BNP	LEOS	MAXIMUM TRIES DONE, ERROR
	STH	R14.HALFWORD	SAVE VALUE
	B	EZASEND	GO SEND AGAIN
SENTALL	L	R14.DSR14SAV	RESTORE RETURN REGISTER
	BR	R14	RETURN
READ RESPONSE FROM MAIL HOST			
VL, MF=(E,PARMLIST)			
SPACE			
AFTSELT	EQU		
	CLC	RSNDMSK,RRETMASK	CHECK READ RETURN MASK
	BE	READCALL	GOT SOMETHING, GO GET IT
	CLC	RETCODE,=F'0'	CHECK RETURN CODE
	BH	READCALL	GO READ SOCKET
	CLC	RETCODE,=F'-1'	IS IT NEGATIVE 1
	BE	LEOS	YES, MUST BE A TCP/IP ERROR
	B	NORESPN	
READCALL	MVC	SOCKFUNC,=CL16'READ'	SOCKET FUNCTION = READ
	XC	INBUF,INBUF	CLEAR BUFF
	MVC	NBTES,=ALL(L'INBUF)	LENGTH TO READ
SPACE			
	CALL	EZASOKET,(SOCFUNC,SOCRECV, NBYTES,INBUF,ERRNO,RETCODE), VL, MF=(E,PARMLIST)	
SPACE			
	CLC	RETCODE,=F'0'	DID WE GET ANYTHING?
	BE	NORESPN	NO, MUST NOT BE SENDING ANYTHING
	BH	TRANRESP	YES, GO TRANSLATE TO EBCDIC
	CLC	RETCODE,=F'-1'	IS IT NEGATIVE 1
	BE	LEOS	YES, MUST BE A TCP/IP ERROR
	B	NORESPN	
TRANRESP	EQU	*	
	CALL	EZACICOS,(INBUF,RETCODE),VL, MF=(EPARMLIST) TRAN. EBCDIC	
AFTRANS	EQU		
	BE	LEOS	YES, MUST BE A TCP/IP ERROR
	B	NORESPN	
TRANRES	CALL	EZACICOS,((R12),RETCODE),VL, MF=(E,PARMLIST) TRAN. EBCDIC	
AFTRAN	EQU		
	MVC	BUF.0(R12)	
NORESP	EQU		
	MVI	NORESPND,C'Y'	
HOSTREX	L	R14.DSR14SAV	RESTORE RETURN REGISTER
	BR	R14	RETURN
ERROR ROUTINES			
LEOM	DS	OH	
	EXEC	CICS ABEND, ABCODE('LEOM'), CANCEL	ABEND THIS TASK
LEOC	EQU	*	
GMAINERR	EQU	*	
	MVC	LEM010RC,LEM010SG	GETMAIN ERROR

	DC	CL40'Subject:	Mail from th system'
Subject:	EQU	(*-HEADER)	
	DC	CL40'	
	DC	CL15'To:	
DEST2@	EQU	(*-HEADER)	
	DC	CL65'	
*	DC	CL80'	
HDR#	EQU	(*-HEADER)/80	
ENDER	DC	CL80'.'	
	DC	CL80'QUIT'	
	DC	CL80'//'	
ENDERS	EQU	(*-ENDER)	
ENDER#	EQU	(*-ENDER)/80	
*CRTBL	DC	25X'00'	
*	ORG	CRTBL+13	
*	DC	X'0D' <CR> TO FIND	
*	ORG	CRTBL+255	
*			
MAX_SECONDS	DC	F'00000005'	
FIONBIO	DC	XL4'8004A77E'	
SO_REUSEADDR	DC	F'00000004'	
SO_KEEPALIVE	DC	F'00000008'	
SO_BROADCAST	DC	F'00000032'	
SO_LINGER	DC	F'00000128'	
SO_OOBINLINE	DC	F'00000256'	
SO_SNDBUF	DC	F'00004097'	
SO_ERROR	DC	F'00004103'	
SO_TYPE	DC	F'00004104'	
DOWTABLE	DS	OD	
	DC	XL4'0',C'Sun,'	
	DC	XL4'1',C'mon,'	
	DC	XL4'2',C'Tue,'	
	DC	XL4'3',C'Wed,'	
	DC	XL4'4',C'Thur,'	
	DC	XL4'5',C'Fri,'	
	DC	XL4'6',C'Sat,'	
SATURDAY	DC	XL4'6',C'Sat,'	
DOW\$	EQU	*-SATURDAY	
DOW#	EQU	(*-DOWTABLE)/DOW\$	
MONTABLE	DS	OD	
	DC	XL4'1',C'Jan'	
	DC	XL4'2',C'Feb'	
	DC	XL4'3',C'Mar'	
	DC	XL4'4',C'Apr'	
	DC	XL4'5',C'May'	
	DC	XL4'6',C'Jun'	
	DC	XL4'7',C'Jul'	
	DC	XL4'8',C'Aug'	
	DC	XL4'9',C'Sep'	
	DC	XL4'A',C'Oct'	
	DC	XL4'B',C'Nov'	
DECEMBER	DC	XL4'C',C'Dec'	
MON\$	EQU	*-DECEMBER	
MON#	EQU	(*MONTABLE)/MON\$	
*SSTDDATA	DS	OH	
*			
CTLBLOCK	DC	CL(TCPBLEN)'	
*	DC	CL8'	
	DC	CL6,000010'	

```

SPROCESA  DC  CL8'PROCESS'
           DC  CL10'VERIFY'

           DC  CL8'
           DC  CL6'000016'
           DC  CL8'STATEKEY'
SSTATEKA  DC  CL16'XXXXXXXXXXXX'

           DC  CL8'
           DC  CL8'
           DC  CL6'000015'
           DC  CL8'IPADDR'
SIPADDRA  DC  CL15'128.227.72.2

           DC  CL8'
           DC  CL6'000005'
           DC  CL8'PORTNUMB'
SPORTNMA  DC  CL5'03005'

           DC  CL8'
           DC  CL6'000020'
           DC  CL8'STUID'
SSTUIDA   DC  CL20'

           DC  CL8'
           DC  CL6'000015'
           DC  CL8'PASSWORD'
SPASSWDA  DC  CL15'

           DC  CL8'
           DC  CL6'000001'
           DC  CL8'RETNCODE'
RRETNCA   EQU  *-RSTDDATA
           DC  CL5'

           DC  CL8'
           DC  CL6'000080'
           DC  CL8'RETNMESS'
RRETNMA   EQU  *-RSTDDATA
           DC  CL80'
RDATALEN  EQU  *-RSTDDATA

END      RAMI1S00

```

APPENDIX H

DATA SET RAMI00

1. RAMI0400 FILE RECORD SECT

ALL INFORMATION IN THIS DOCUMENT, INCLUDING THE TRANSPORT MECHANISM AND PROTOCOLS, ARE COPYRIGHT 1998 BY THE UNIVERSITY OF FLORIDA. ALL RIGHTS RESERVED.

COPY RARARGDA
USING RARARGDA,R2

COPY RARAMMEN
USING REARAMMEN,R12
COPY RAMICOMM
USING COMMAREA,R5
COPY RARATCPB

COY RARAEXLG

2. WORKING STORAGE

COPY DFHAID 3270 AID CHARACTERS

DFHEISTG	COPY RAREGS	REGISTER EQUATES
	EJECT	
	DSECT	
	SCRATCH PAD AREA	
	COPY RARASPAD	
	FIELDS FOR RAMIGETD	
FINDFLD	DS CL8	
FINDFLDT	DS CL79	
UPDATE	DS C	

SAVER12	DS	F
TEMP9 DS	DS	CL6
TEMP06	DS	CL6
RGFLUX	DS	CL2
RGAPPT	DS	C
RGDAYO	DS	C
PHASE	DS	C
BADSSNF	DS	C
NOSSNF	DS	C
CREDITS	DS	PL2
CTLKEY	DS	CL3
APPT	DS	CL2
TIMEAP	DS	CL2
DADATE	DS	0CL9
DADUMM	DS	CL1
DAMMDD	DS	0CL4
DAMM	DS	CL2
DADD	DS	CL2
DATEIME	DS	CL4
TIMEALL	DS	OCL5 BHHMM
TIMEBLNK	DS	C
TIMEHH	DS	CL2
TIMEMM	DS	CL2
SYSSMMDD	DS	CL4
PTIME	DS	PL4
PSTIME	DS	PL4

ADD24	DS	PL4
PRTIME	DS	PL4
SYSTIME	DS	OCL6
SYSHHMM	DS	OCL4
SYSHH	DS	CL2
SYSMM	DS	CL2
SYSSS	DS	CL2
REGMTH	DS	CL3
REGDATE	DS	OCL4
REGMM	DS	CL2
REGDD	DS	CL2
REGTIME	DS	CL4
PACK3	DS	PL3
DIVIDEND	DS	OD
QUOTIENT	DS	F
REMAINS	DS	F
DIVISOR	DS	F
SAVETRID	DS	F
SAVER6	DS	F
SAVESEL	DS	F
SAVEMVAR	DS	F
SAVEMVLN	DS	H
TEMPAREA	DS	CL30
SAVESTID	DS	CL20
SAVEPSWD	DS	CL10
SAVEPASS	DS	CL4
EJECT		

PROGRAM RAMI0400 STARTS HERE

RAMI0400	DFHEIENT	CODERE=(3,8)
RAMI0400	AMODE	31
RAMI0400	RMODE	ANY
L	R5,DFHEICAP	
CLC	EIBCALEN,=H'0'	
BE	RETURN	
EXEC	CICS HANDLE CONDITION DISIDERR(ERROR)	
	ERROR(ERROR) TERMIDERR(ERROR) SYSIDERR(ERROR)	
	ISCVINREQ(ERROR) INVREQ(ERROR) IOERR(ERROR)	
	DISABLED(ERROR) ILLOGIC(ERROR) LENGERR(ERROR)	
	NOTAUTH(ERROR)	

START	EXEC	CICS HANDLE CONDITION NOTFND(NOTEND) NOTOPEN(NOTOPEN)	
	EQU	*	
	XC	NUMCHEC,NUMCHECK	
	MVC	COMAFNDF,=C'MDASPSWD'	DATA IDENTIFIER
	BAL	R4,GETDATA	GO GET IT
	XC	SAVEPSWD,SAVEPSWD	INIT FIELD
	LH	R14,=H'10'	LENGTH OF DATA
	LA	R15,SAVEPSWD	PUT DATA HERE
MVC	COMAFNDF,=CMDASPSW2'	DATA IDENTIFIER	
BAL	R4,GETDATA	GO GET IT	
CLC	SAVEPSWD,BLANKS		
BNH	NO2ND		
CLC	SAVEPSWD,COMASELC		
BNE	NOTSAME		
MVI	COMASELC+4,C','		
MVC	COMASELC+5(4),SAVEPSWD		

```

NO2ND      MVC  SAVESELC,COMASELC
            MVC  COMAIBFN,BLANKS

            EXEC CICS LINK PROGRAM('RAMI1S00') COMMAREA(COMMAREA)
                  LENGTH(COMACOML)
            CLI  COMAERRF,C'Y'      SUCCESSFUL READ?
            BE   RETURN             NO, GET OUT

            CLC  COMAIBFN,BLANKS    SUCCESSFUL RETURN
            BH   GOBACK            NO, GET OUT

            B    GETRCOD

            MVC  COMNAIBFN,=C'8'
            B    PUTERROR
NOTSAME     EQU  *
            MVC  COMAIBFN,=C'10'
            B    RETURN

BAD         EQU  *
            MVC  COMAIBFN,=C'6'
            B    PUTERROR
IBFN16      EQU  *
            MVC  COMAIBFN,=C'16'
            B    PUTERROR
            EQU  *
            MVC  COMAIBFN,=C'3'
            MVC  COMAPASS,SAVESELC
            B    RETURN

FORCE        EQU  *
            EQU  *
NOTOPEN      EQU  *
NOTFND       EQU  *
            B    PUTERROR
PUTERROR     EQU  *
            MVC  COMAMESS(8),=C'CONTINUE'
                  MVI  COMAERRF,C'Y'
                  B    RETURN
PINOK        EQU  *
            MVC  COMAVUID(9),COMAEXSS
            MVC  COMATOPC,COMATARG
            MVC  COMAEXSS+3(6),SAVESTID+13

            CLC  COMAEXSS,BLANKS
            BH   CKSSNNUM
            MVI  NOSSNF,C'Y'
            B    READMMEN

CKSSNNUM    MVZ  NUMCHEC,COMAEXSS
            CLC  NUMCHEC(9),NUMZONE
            BE   SSNOK
            MVI  BAEDSSNF,C'Y'
            B    READMMEN

SSNOK        EQU  *
            MVC  COMASELC(4),SAVESTID+3

CKFORCE      CLC  COMADATA(5),=C'FORCE'
            BNE  DOPIN

            MVI  COMASELC+4,C'
            MVC  COMASELC+5(4),SAVESTID+3

```

```

LH    R14,=H'4'          LENGTH OF DATA
LA    R15,SAVEPASS        PUT DATA HERE
MVC   COMAFNDF,=C'PASSWORD' DATA IDENTIFIER
BAL   R4,GETDATA         GO GET IT
CLC   SAVEPASS,BLANKS
BNH   DOPIN
MVC   COMPASS,SAVEPASS

DO PIN ROUTINE
DOPIN  EQU  *
MVC   SAVETRID,EIBTRNID
MVC   IEBTRNID,=C'RA56'
MVC   COMASAVP,=C'RAMI0500'
EXEC CICS LINK PROGRAM('RAGA3000') COMMAREA(COMMAREA)
LENGTH(COMACOML)
MVC   EIBTRNID,SAVETRID

CLC   COMASELC(5)M=C'+0050      EXPIRED?
BNE   PUTFIX
MVI   COMASELC+4MC'1'          EXPIRED?

PUTFIX  MVI   COMASELC,C'0'          ←-----TEMP FIX!!!!!
MVC   COMARCOD(5)MCINASELC
MVC   COMARMES,BLANKS
*
B    RETURN
*
*
*GOBACK  EQU  *
RETURN   EXEC CICS RETURN
*
PUTDATAE  MVC   0((*-*,R12),0(R6)
BUMPADD   LA    R12,(*-*)(R12)
*
NUMZONE  COPY  RAMIGETP
BLANKS    DC    12C'0'          ZONES
MAXWORK   DC    CL120'
DC    HL2'2000;
SPACE
EJECT
LTORG   RETURN TO CICS

END   RAMI0400

```

EROBBIN
S3030

JOBID	TSU09089
JOB NAME:	EROBBIN
UERID:	EROBBIN
SYSOUT CLASS:	A
OUTPUT GROUP:	4 .00001.00001
TITLE:	
DESTINATION:	UFG3820
NAME:	ROBBINS, EARL

ROOM:
 BUILDING:
 DEPARTMENT:
 ADDRESS:

PRINT TIME: 11:37:13 AM
 PRINT DATE: 11 AUG 1998
 PRINTER: PRT6
 SYSTEM ID: NER1

EROBBIN
 S303

JOBID: TSU0989
 JOB NAME: EROBBIN
 USERID: EROBBIN
 SYSOUT CLASS: A
 OUTPUT GROUP: 5
 TITLE:
 DESTINATION: UFG3820
 NAME: ROBBINS, EARL
 ROOM
 BUILDING:
 DEPARTMENT:
 ADDRESS:

PRINT TIME: 11:37:24 AM
 PRINT DATE: 11 AUG 1998
 PRINTER: PRT6
 SYSTEM ID: NER1

TSO FOREGROUND HARDCOPY

DSNAME=RA.PATENT (RAMI0600)

DATA SET RAMI0600

00010000

00020000

ALL INFORMATION IN THIS DOCUMENT, INCLUDING THE TRANSPORT
 MECHANISM AND PROTOCOLS, ARE COPYRIGHT 1998 BY THE
 UNIVERSITY OF FLORIDA. ALL RIGHTS RESERVED.

00030006

00030106

00030206

00030306

00030406

COPY RARAMDAS
 USING RARAMDAS,R9
 COPY RARARGDA
 USING RARARGDA,R12
 SPACE

00031000

00040000

00041000

00042000

00050000

00060000

00070000

00080000

2. WORKING STORAGE

DFHEISTG	COPY DFHAID SPACE DSECT	3270 AID CHARACTERS REGISTER EQUATES COMMAREA DSECT
	SCRATCH PAD AREA	00090000 00100000 00110000 00120000 00130000 00140000
TMPKEY	DS OCL16	00150000
TMPTIME	DS PL8	00170000

TMPMDASK	DS	CL8	00170100
LINE	DS	CL80	00170200
MIDASSTE	DS	CL8	00170300
COUNT	DS	PL3	00171000
ABCODE	DS	CL4	00171100
	COPY	RARASPAD	00172000
	EJECT		00201200
			00201301
			00201700
			00201800
RAMI0600	DFHIENT	CODEREG=(3,8)	00201900
RAMI0600	AMODE 31		00202000
RAMI0600	RMODE ANY		002030000
	SPACE		00203100
		REQUIREMENT CHECKING AND EVLAUATION ROUTINES	00203200
			00204000
			00205000
			00206000
	EXEC CICS HANDLE CONDITION DSIDERR(ERROR)		00207000
	ERROR(ERROR) TERMIDERR(TRMIDERR)SYSIDERR(ERROR)		00208000
	ISCVREQ(ERROR) INVREQ(ERROR) IOERR(ERROR)		00208100
	DISABLED(ERROR) ILLOGIC(ERROR) LENGERR(ERROR)		00208200
	NOTAUTH(ERROR) DUPREC(ERROR)		00208300
	EXEC CICS HANDLE ABEND LABEL (HNDABND)		00208400
	EXEC CICS HANDLE CONDITION NOTOPEN(NOTOPEN)		00286000
	MVC MDASKEYYY,TMPMDASK		00287000
	MVT MDASKEYYY,C'0'		00208900
	00469700		
	EXEC CICS WRITE FILE('RAMDAS') FROM(RARAMDAS)		00469800
	LENGTH(HALFWORD) RIDFLD(MDASKEYYY)		00469900
			00470000
AP	COUNT,=P'1'		00470100
			00470200
	---DEQ RESOURCE---		00470300
			00470400
	EXEC CICS DEQ RESOURCE(MIDASSTE) LENGTH(8)		00470500
	B NEXT		00470600
			00470900
ERROR	EQU *		00471000
MVC	LINE(41),=C'THERE WAS AN ERROR PROCESSING THE REQUEST		00551000
B	SENDLIN		00552000
TRMIDERR	EQU		00553000
B	RETURN		00553100
			00553300
NOTOPEN	EQU		00554000
MVC	LINE(38)=0TATE FILE OR ALTERNATE INDEX NOT OPEN		00555000
B	SENDLINE		00556000
NOTFND	EQU *		00560000
MVC	LINE(30),=C'REQUESTED RECORD WAS NOT FOUND'		00560100
B	SENDLINE		00560200
STARTAGN	EQU *		00560300
	EXEC CICS START TRANSID('MI06'		00560400
ALLDONE	EQU		00560500
MVC	LINE,MESSMI06		00560600
UNPK	LINE+39(3),COUNT		00560700
OI	LINE+41,C'0'		00560800
			00560900
SENDLINE	EQU		00561002
EXEC	CICS HANDLE CONDITION INVREQ(RETURN) NOTALLOC(RETURN)		00561103

	EXEC CICS SEND FROM(LINE)	00561202
RETURN	EQU *	00561402
	EXEC CICS DEQ RESOURCE(MIDASSTE) LENGTH(8) NOHANDLE	00561502
		00561602
RETURN2	EQU *	00561804
HNDABND	EQU *	00561902
	EXEC CICS ASSIGN ABCODE(ABCODE)	00562002
	EXEC CICS ABEND ABCODE(ABOCED) NODUMP	00562102
	B RETURN	00562202
		00562301
		00562401
		00562801
BLANKS	DC CL80'	00563401
MESSMI06DCCL80'MI06 COMPLETED SUCCESSFULLY, PROCESSED RECORDS		00563500
		00563600
		00563700
	LTORG	00564000
	END RAMI0600	00565000
		00565100
		00566000

APPENDIX J

nirvana.c—EAGLE transfer client for Web server.
Written by Michael Lucas for the University of Florida
November 11, 1998 <http://www.cs.ugr.es/~mjl/nirvana/nirvana.c>

Incorporates client.c by Tom Kelliher and getcgivars.c from NCSA

Ver 1.1.6 build 0 Robustified(tm)!

Program flow

This program is started by a call from a Web page which contains a form. Data is passed via a CGI string which is then parsed via the standard NCSA getcgivars.c program into a two-dimensional array of strings: name of the field paired with the value for the field. (Note that the program must be compiled in a directory where getcgivars.c exists, or is in the path.)

This program then sends a message to the appropriate CICS server and port (both defined near the top of this program) to start the transaction M100.

When CICS receives this call it responds with OKTOSEND to indicate readiness to receive data. This program then sends all the CGI data in the form

8-byte-field-name field-data carriage-return
without the spaces (eg: MYFIELDNTTHISISTHEDATA\n).

When CICS has received the data it then does magic and returns a 50-byte data descriptor and a fully-marked-up Web page. This program parses the stream and writes the page to standard out, where it is picked up by the CGI form and sent as a Web page to the browser that originally called it.

RECENT CHANGE LIST

11/03/1998 ver 1.1.4 "Robustified!"

Repaired code that sent garbage to Web browser on dead socket call, causing random text boxes or garbage to appear

APPENDIX J

11/17/1998 ver 1.1.5

Added alarms piped to pdie, to prevent child processes forked by the SP from running forever if they fail to receive the needed information from the data client (e.g. CICS abend or online application hork by an AOL user).

11/19/1998 ver 1.1.6

Modified die and pdie to accept the socket address so they canclose the socket on err. Modified all calling functions pass sock to die and pdie.

```
#include <sys/types.h>
#include <sys/socket.h>
#include <netinet/in.h>
#include <netdb.h>
#include <stdio.h>
```

```

#include <stdlib.h>
#include <signal.h>
#include "getgivar.c"

#ifndef      prod
#define SERVER_PORT 9999           production port
#else
#define SERVER_PORT 9999           test port
#endif

#define BUFFER_SIZE 10240          Input info string
#define MAX_STRING_LENGTH 20480
#define CTL_BUFFER_SIZE 60          Output info string
#define UF_CICS "mvs.your.site"    CICS server
#define NUMBER_OF_XFER_ATTEMPTS 5
#define TRANS_TYPE "M100"
#define DIAG_MSG_SIZE 8
#define OK_MSG "GOTDATAOK"
#define SPACES " "


prototypes

void die (const char *, int);
void die (const char *, int);

void print_top(void);
void zero_pad(int, char *);

main

int main (int argc, char *argv[])
{
    int sock;                      fd for socket connection
    struct sockaddr_in server;      Socket info. for server
    struct sockaddr_in client;      Socket info. about us
    in clientLen;                  Length of client socket struct.
    struct hostent *hp;             Return value from gethostbyname()
    char buf [BUFFER_SIZE];        Received data buffer
    char html_out[MAX_STRING_LENGTH]; Build string for output
    register int l;                 loop counter
    int byte_out = 0;               Total bytes sent on a socket write
    int byte_in = 0;                Total bytes received on a socket read
()
    int count = 1;                  miscellaneous counter
    int byte_tot = 0;
    register int rec_size = 0;
    register int rec_size = 0;
    char **cgivars;
    char outbut[CTL_BUFFER_SIZE];   Data written to the socket
    char outbuf2[BUFFER_SIZE];      Data written to the socket
    char zero_stg[6];
    int var_size = 0;                Holds the length of the cgi variable
    char holder[25];
}

```

Install the signalling system to abort processing in case of error.

```

signal(SIGALRM, die("No response from data client. Nirvana terminating.",
s ck));

```

```

alarm(600);

Parse the cgi string and pull out a 2d array of data
cgivars = getcgivars();

Open a TCP socket and send message.
if ((sock = socket (AF_INET, SOCK_STREAM, 0)) < 0)
    pdie("Couldn't open stream socket",sock);

Prepare to connect to server.

bzero((char *) &server, sizeof(server));
server.sin_family =AF_INET;
if ((hp = gethostbyname(UF_CICS)) == NULL)

{
    printf("%s: unknown host\n", UF_CICS);
    die(buf, sock);
}
bcopy(hp->h_addr,&server.sin_addr, hp->h_length);
server.sin_port = htons((u_short) SERVER_PORT);

Try to connect.

if (connect(sock, (struct sockaddr*) &server, sizeof(server)) ,0)
    pdie("Couldn't connect stream socket", sock);

Determine what prot client's using.

clientLen = sizeof(client);
if(getsockname() overwrote name structure", sock);

Prepare to send to CICS to kick off the transaction.

if (byte_out = write(sock, TRANS_TYPE, sizeof(TRANS_TYPE)) < 0 )
    pdie("Couldn't write TRANS_TYPE on stream socket", sock);

Check that CICS is happy and ready to receive data.

if (byte_in = read(sock, buf, DIAG_MSG_SIZE))           If we got data
{
    if(stemp(buf,"OKTOSEND"))                         Check for ready
        die("CICS did not reply ready.", sock);
    These brackets are needed to define the scope of the IF ELSE
    condition.

else
    pdie("No answer from socket reading ready message from CICS.",

Start formatting the HTML page.

print_top();

Flush the strings.

bzero(html_out, sizeof(html_out));
bzero(outbuf2, sizeof(outbuf2));

```

Copy the Web page variables to the output string in this format:

8 spaces, variable length(6), variable name, variable data, newline.
Copy until out of data.

```
for (l=0; cgivars[l]; l++)
{
    bzero(holder, sizeof(holder));

    Pull any button data

    if (! stemp (cgivars[l], "MIDASBTN"))
        l += 2;
    var_size = strlen (cgivars[l + 1]);
    zero_pad(var_size, zero_stg);
    strcat(outbuf2, zero_stg);
    sprint(holder, "%d", var_size);
    strcat(outbuf2, holder);
    strcat(outbuf2, cgivars[l]);
    strcat(outbuf2, cgivars[l + 1]);
    strcat(outbuf2, "\n");
}
```

Move the length info to the beginning of the output stream.
It will be the length of outbuf PLUS the information itself.

```
rec_size = (CTL_BUFFER_SIZE + strlen(outbuf2));
```

```
zero_pad)rec_size, zero_stg);
```

```
sprintf(html_out,"%s%d", zero_stg_rec_size);
```

```
strcat(html_out,"012345");
```

Add the output buffer to the transmission string.

```
strcat(html_out, outbuf2);
```

Write html_out to the socket.

```
count = 1;
byte_tot = 0;
while (byte_tot < rec_size)
{
    if ( ! (byte_out = write(sock,html_out, rec_size)))
        count++;
    if (byte_out < 0)
        pdie("Exceeded number of write attempts on socket", sock);
    byte_tot += byte_out;
}
```

Clear all buffers and counters to prepare to read CICS response.

```

count = 1;
rec_siz = 0;
byte_tot = 0;
bzero(buf, sizeof(buf));
bzero(outbuf2, sizeof(html_out));

```

Read the CICS response from the socket.

```

if ( ! (byte_in = read(sock, buf, CTL_BUFFER_SIZE)))
{
    If null read, increment bad read counter and test for excess.
    count++;
    if (count == NUMBER_OF_XFER_ATTEMPTS)
        pdie("Excess number of failed reads on initial socket read. Check

```

```
communications", sock);
    }
else    We got data on the read, continue
{
    Move the first 6 bytes of buf into outbuf2
```

```

    strncpy(outbuf2,buf,6);
    rec_size = atoi(outbuf2);
    byte_tot = rec_size - byte_in;

```

```

    if (byte_in < 0)
        pdie(buf, sock);

```

Skip the control info and copy the remainder to the Web output buffer.

```

    for(i = 0; i < byte_in; i++)
        printf("%c", buf[(i + CTL_BUFFER_SIZE)]);
    }

```

If we didn't get the entire stream in the first pass, loop until we do.

```

while (byte_tot > 0)
{
    bzero(buf, sizeof(buf));
    byte_in = read(sock, buf, byte_tot);
    if (byte_in < 0)
        pdie("Died while in the socket reading loop", sock);
    if(count == NUMBER_OF_XFER_ATTEMPTS)
        pdie("Exceeded number of dead reads on loop socket read",
    sock);
    if(byte_in == 0)           If dead read on socket
        count++;           increment bad-read counter
    }
else
{
    byte_tot -= byte_in;
}

```

```

        printf("%s", buf);      Send buf to webs rver
        count = 0;             Reset bad-read counter
    }

Send confirmation back to CICS

if(byte_tot <= rec_size)      which will be true if
{                            we got the data
    if( ! (byte_out = write(sock, OK_MSG, sizeof(OK_MSG))))
        pdie("Died while confirming transmission received",sock);
}

Write out the end-html string to the Web server

printf("</body></html>");

Close this connection.

close(sock);

}

End of main

```

pdie --- Call perror() to figure out what's going on and die.

```

void pdie(const char *mesg, int sock)
{
    printf("<docy bgcolor=cc0000 text = fffff><title>Error Report</title>\n");
    printf("<h3><center>Application Error Report </h3><hr>\n");
    printf("<br><i>Problem: </i> %s. Check your data and try
reloading. <br>\n",mesg);
    perror(mesg);
    printf("<hr></center></body></html>");
    close(sock);
    exit(1);
}

```

die --- Print a message and die.

```

void die(const char *mesg, int sock)
{
    printf("<br><i>Error! : %. Try reloading, or contact support.\n",mesg);
    fputs(mesg, stderr);
    fputc('\n', stderr);
    close(sock);
    exit(1);
}

```

print_top --- Start output of the HTML page

```
void print_top(void)
{
    printf("Content-type: text/html\n\n");
    printf("<html>\n");
    printf("<! All information in this document, including the transport>\n");
    printf("<! mechanism and protocols, are copyright 1998 by the >\n");
    printf("<! University of Florida. All rights reserved. >\n");
```

Uncomment the following line to enable no-cache for production.

```
printf(",META HTTP-EQUIV = \"Pragma\"Content=\"no-cache\">\n");
```

zero_pad --- Adds the leading zeros to the data length description

```
void zero_pad(int rec_size, char *zero_stg)
{
    bzero(zero_stg, sizeof(zero_stg));

    if (rec_size <= 9)
        strcpy(zero_stg, "00000");
    else if (rec_size <=99)

        strcpy(zero_stg, "0000");
    else if (rec_size <=999)
        strcpy(zero_stg, "000");
    else if (rec_size < 9999)
        strcpy(zero_stg, "00");
    else if (rec_size <= 9999)
        strcpy(zero_stg, "0");
```

What is claimed is:

1. A method of providing Web access to data, the steps comprising:

providing a Web server for distribution of data to users;

providing a database on a database computer operably connected to the Web server, the database having a database program that provides data in a given format;

upon a user requesting data in the database, the Web server forwarding the request to the database, the data base program accessing the data in the given format, the database computer running a Web control program that generates a Web page with the requested data and supplies the generated Web page to the user.

2. The method of Claim 1 wherein the Web control program marries file definition objects and page definition objects to generate Web pages.

3. The method of Claim 1 wherein the Web server is on a Web server computer different from the database computer.

4. The method of Claim 3 wherein the database computer is a mainframe computer.

5. The method of Claim 4 wherein the mainframe computer supplies the generated Web page to the user via the Web server computer.

6. The method of Claim 4 wherein, upon a user requesting data not in the mainframe computer, the Web control program causes the mainframe computer to access data on a remote computer and the Web control program then generates a Web page with the requested data and supplies the generated Web page to the user.

7. The method of Claim 4 wherein, responsive to a user, the Web control program calls a subroutine on the mainframe computer, which subroutine is independent of the Web control program and is a legacy subroutine.

8. The method of Claim 4 further including the steps of: having a user supply an identification code, forwarding the identification code from the Web server computer to the mainframe computer, the Web control program then generating a state key, incorporating the state key into a Web page supplied to that user, and wherein the mainframe computer detects the state key from the Web page upon the user requesting data and decides whether that user can access the requested data based on the state key.

9. The method of Claim 8 wherein the state key is randomly generated for a given access session and the state key loses its ability to authorize data access if it is not sent by the user to the mainframe computer at least once during a time-out interval.

10. The method of Claim 4 wherein the Web control program receives a page change command from a user using a Web browser and, responsive to the page change command, the Web control program changes a given Web page to a changed Web page such that any users accessing the given Web page now receive the changed Web page, the page change command changing the appearance of a given Web page.

11. The method of Claim 1 further including the steps of: having a user supply an identification code, the Web control program then generating a state key, incorporating the state key into a Web page supplied to that user, and wherein the database computer detects the state key from the Web page upon the user requesting data and decides whether that user can access the requested data based on the state key.

12. The method of Claim 1 wherein the Web control program receives a page change command from a user using a Web browser and, responsive to the page change command, the Web control program changes a given Web page to a

changed Web page such that any users accessing the given Web page now receive the changed Web page, the page change command changing the appearance of a given Web page.

13. A method of providing Web access to data, the steps comprising:

providing a Web server for distribution of data to users;

providing a database on a database computer operably connected to the Web server, the database having a database program that provides data in a given format;

providing a Web control program on the database computer;

having a user supply an identification code to the Web server, the Web control program then generating a state key and incorporating the state key into a Web page supplied to that user, and wherein the database computer detects the state key from the Web page upon the user requesting data and decides whether that user can access the requested data based on the state key.

14. The method of Claim 13 wherein the state key loses its ability to authorize data access if it is not sent by the user to the database computer at least once during a time-out interval.

15. The method of Claim 14 wherein the Web server is on a Web server computer different from the database computer.

16. The method of Claim 15 further including the step of forwarding the identification code from the Web server to the database computer.

17. The method of Claim 16 wherein the database computer is a mainframe computer.

18. The method of Claim 17 wherein the state key is randomly generated for a given access session.

19. The method of Claim 13 wherein the state key is randomly generated for a given access session.

20. A method of managing a Web site, the steps comprising:

providing a Web control program on a computer;
having the Web control program receive a page change command from a user using a Web browser to access the Web site; and,
responsive to the page change command, the Web control program changing a given Web page to a changed Web page such that any users accessing the given Web page now receive the changed Web page, the

page change command changing the appearance of a given Web page.

21. The method of Claim 20 wherein the Web control program marries file definition objects and page definition objects to generate Web pages.

22. The method of Claim 21 wherein the computer having the Web control program is a mainframe computer and users request data on the mainframe computer via a Web server on a Web server computer.

23. The method of Claim 22 wherein the Web control program generates HTML through a completely table driven process, independent of file definitions and page definitions.

24. The method of Claim 23 further including the steps of: having a user supply an identification code, forwarding the identification code from the Web server computer to the mainframe computer, the Web control program then generating a state key, incorporating the state key into a Web page supplied to that user, and wherein the mainframe computer detects the state key from the Web page upon the user requesting data and decides whether that user can access the requested data based on the state key.

102

25. The method of Claim 22 further including the steps of:

providing a database on the mainframe computer, the database having a database program that provides data in a given format;

upon a user requesting data in the database, the Web server forwarding the request to the database, the database program accessing the data in the given format, and, by operation of the Web control program, generating a Web page with the requested data and supplying the generated Web page to the user.

FIG. 1

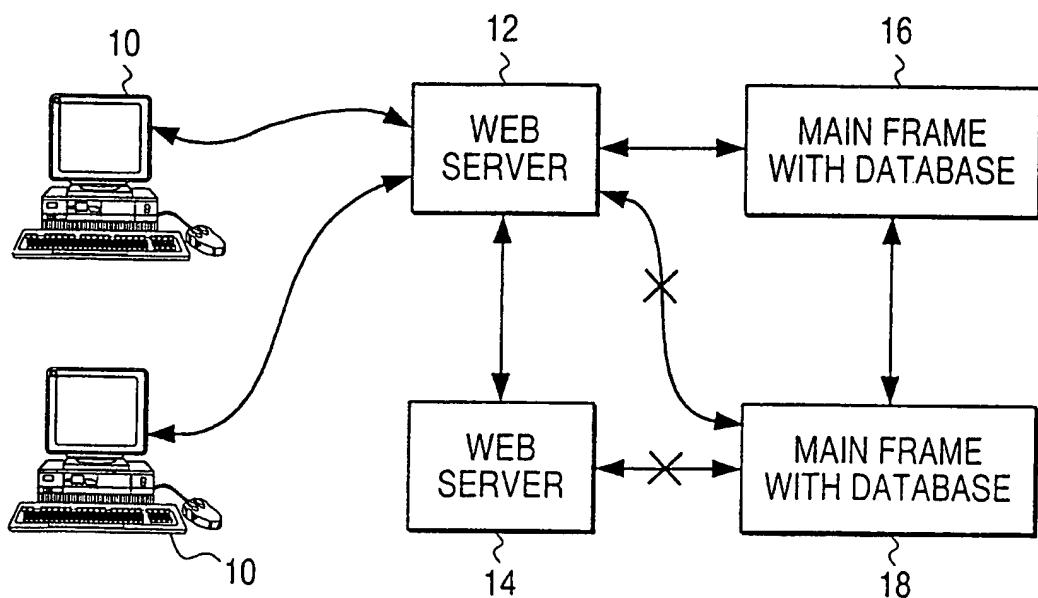
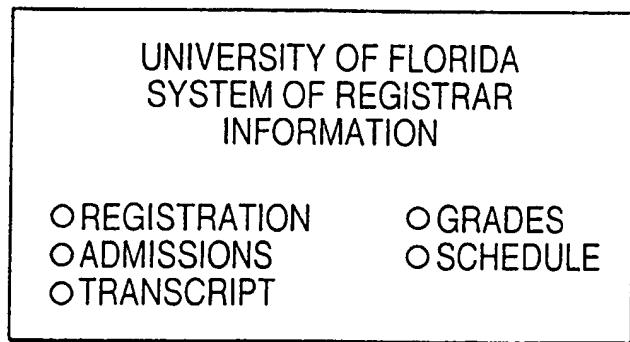


FIG. 2



2/6

FIG. 3

VERIFY STUDENT ID	
ENTER SOCIAL SECURITY NO.	<input type="text"/>
ENTER YOUR PIN	<input type="text"/>

FIG. 4

YOU HAVE GRADES FOR 8 CREDITS		
COURSE	CREDIT	GRADE
AGC2021C	04	A
ECO3100	04	C

HIDDEN STATE KEY

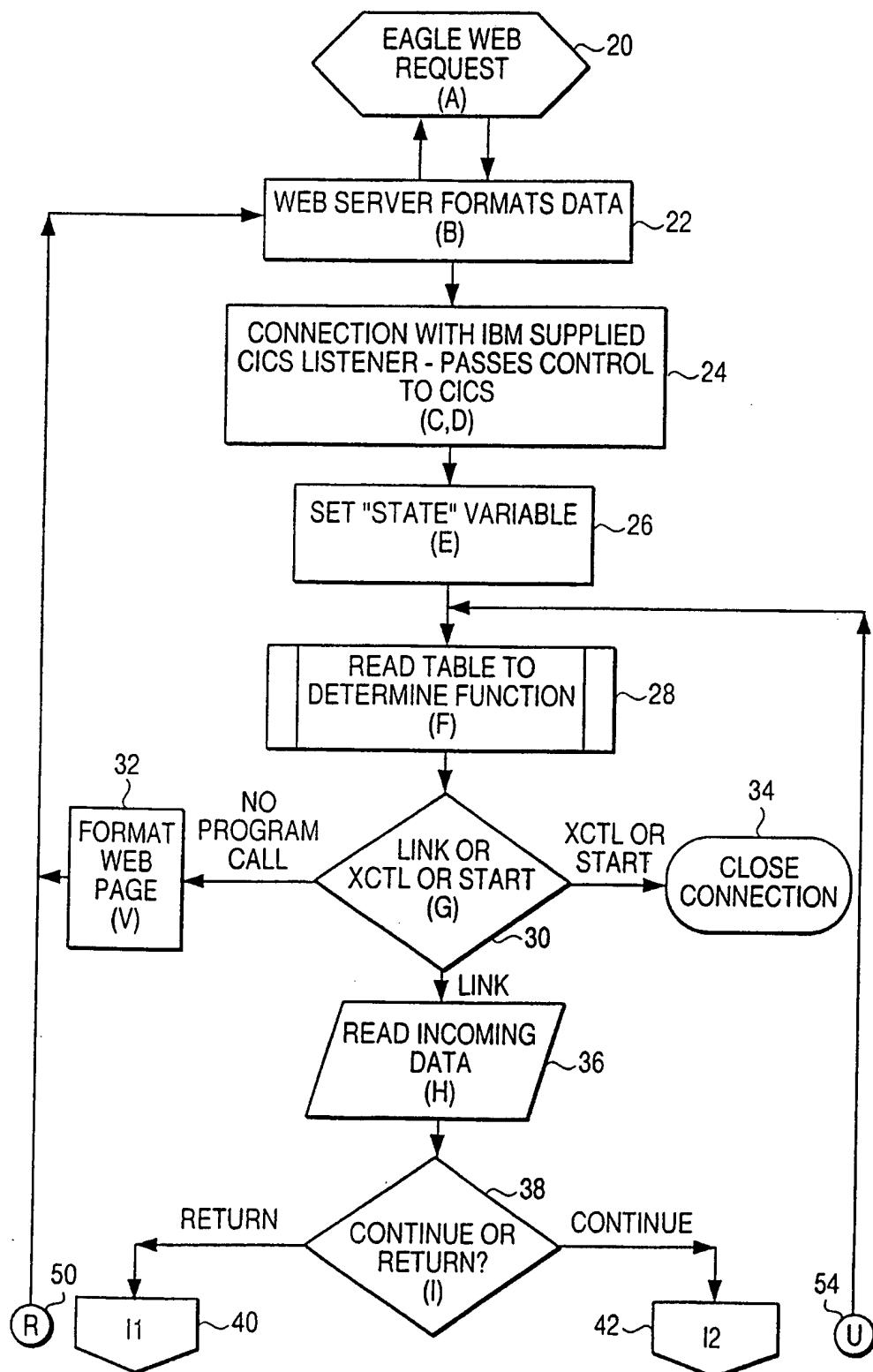
3/6
FIG. 5A

FIG. 5B

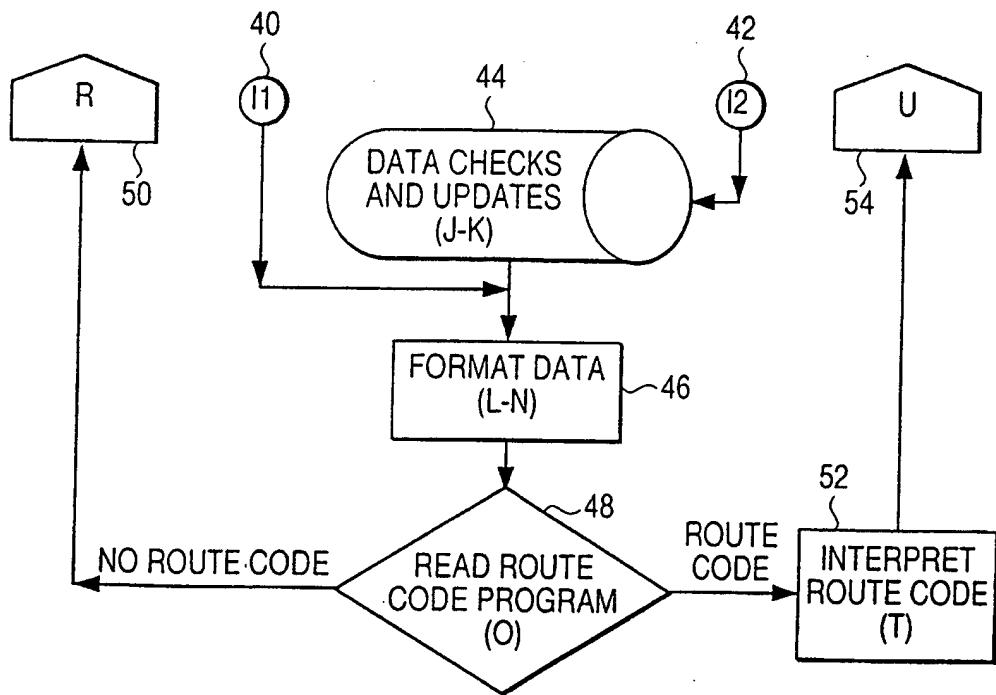
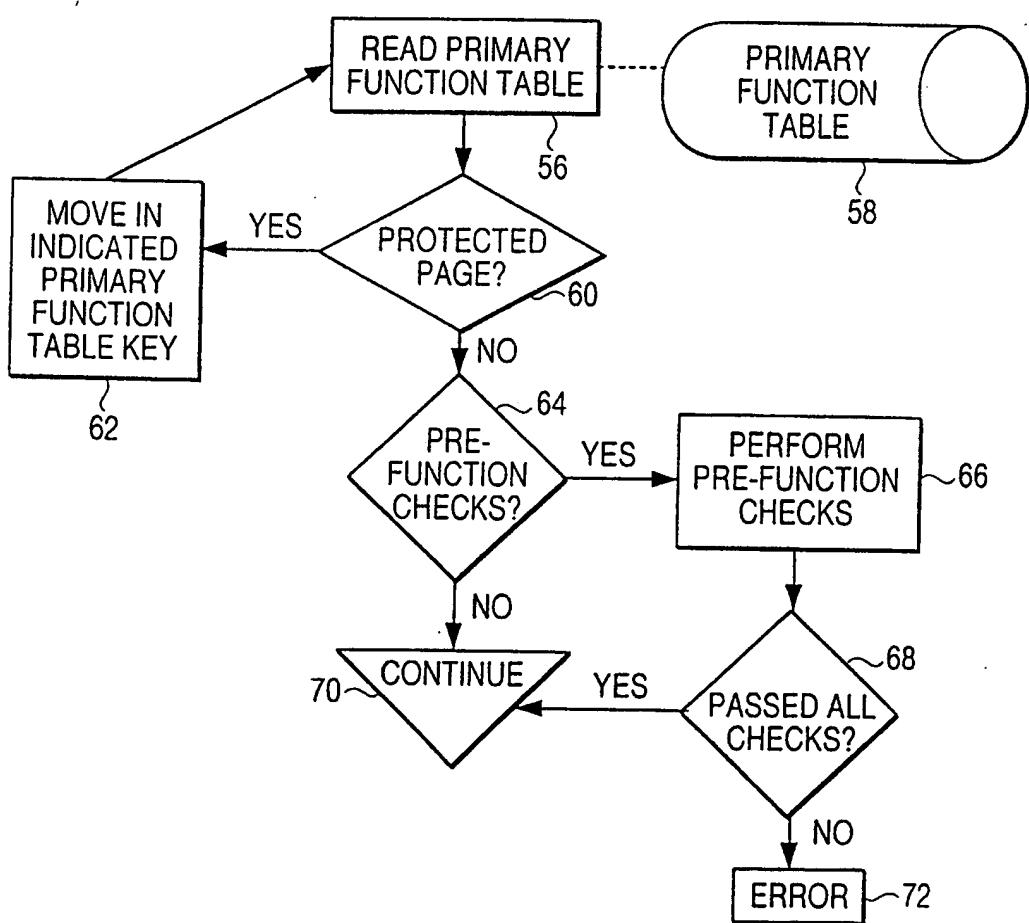


FIG. 6



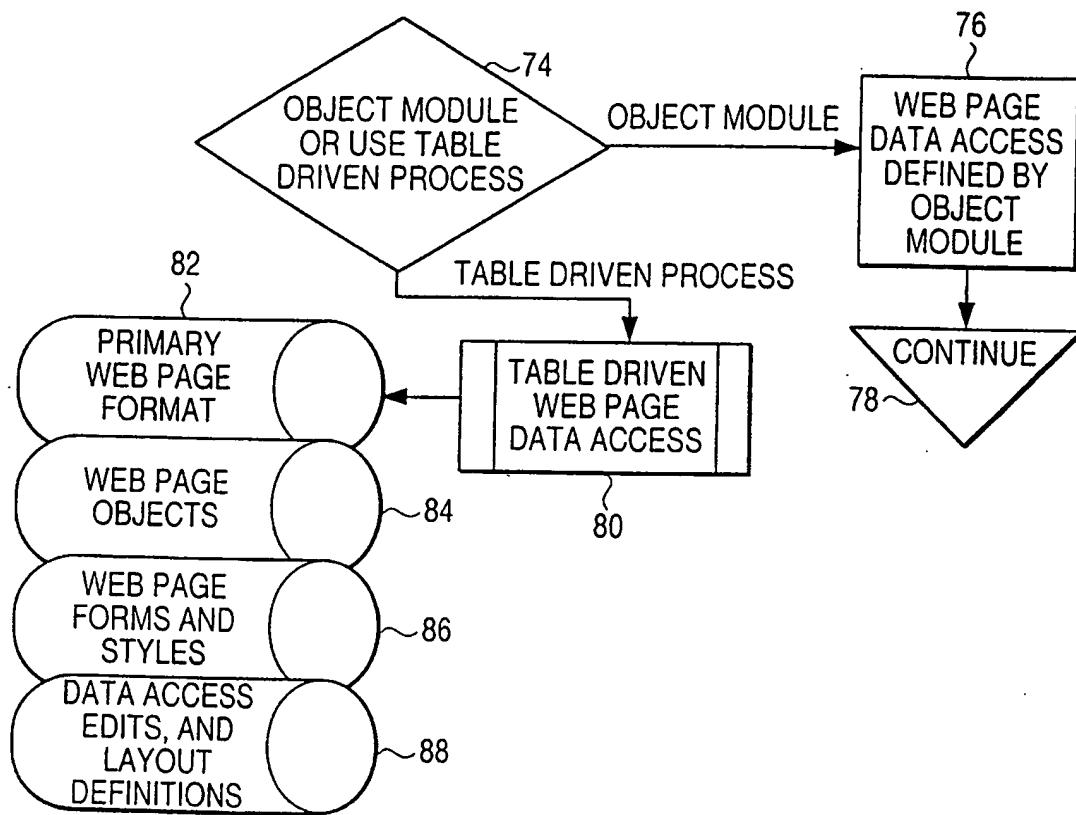
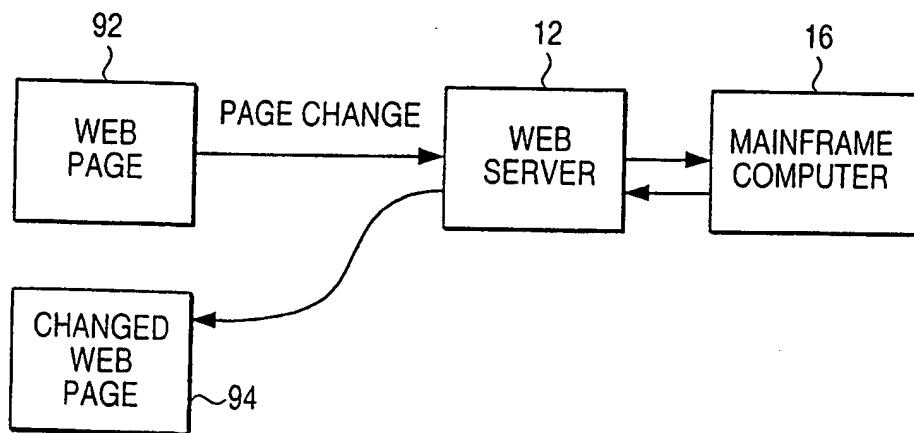
6/6
FIG. 7

FIG. 8



INTERNATIONAL SEARCH REPORT

International application No.
PCT/US99/28321

A. CLASSIFICATION OF SUBJECT MATTER

IPC(7) :G06P 15/00, 17/30

US CL : 707/9, 10; 709/202, 218

According to International Patent Classification (IPC) or to both national classification and IPC

B. FIELDS SEARCHED

Minimum documentation searched (classification system followed by classification symbols)

U.S. : 707/9, 10; 709/202, 218, 200, 203, 217, 219

Documentation searched other than minimum documentation to the extent that such documents are included in the fields searched

Electronic data base consulted during the international search (name of data base and, where practicable, search terms used)

EAST, WEST

C. DOCUMENTS CONSIDERED TO BE RELEVANT

Category*	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
X	US 5,793,964 A (ROGERS et al) 11 August 1998, see Figures 9-11, column 3, line 20- column 7, line 53, column 9, line 26- columns 10, line 43, column 17, line 49- column 18, line 60.	1-25
X, P	US 5,974,441 A (ROGERS et al) 26 October 1999, see the abstract, Figures 9-11, column 19, line 50- column 20, line 62.	1-25
A, P	US 5,926,180 A (SHIMAMURA) 20 July 1999, see the whole document.	1-25
A	US 5,715,453 A (STEWART) 03 February 1998, see the whole document.	1-25

<input type="checkbox"/>	Further documents are listed in the continuation of Box C.	<input type="checkbox"/>	See patent family annex.
--------------------------	--	--------------------------	--------------------------

• Special categories of cited documents.	"T"	later document published after the international filing date or priority date and not in conflict with the application but cited to understand the principle or theory underlying the invention
• "A" document defining the general state of the art which is not considered to be of particular relevance		
• "E" earlier document published on or after the international filing date	"X"	document of particular relevance; the claimed invention cannot be considered novel or cannot be considered to involve an inventive step when the document is taken alone
• "L" document which may throw doubt on priority claim(s) or which is used to establish the publication date of another citation or other special reason (as specified)	"Y"	document of particular relevance; the claimed invention cannot be considered to involve an inventive step when the document is combined with one or more other such documents, such combination being obvious to a person skilled in the art
• "C" document referring to an oral disclosure, use, exhibition or other means	"A"	document member of the same patent family
• "P" document published prior to the international filing date but later than the priority date claimed		

Date of the actual completion of the international search	Date of mailing of the international search report
16 MARCH 2000	12 APR 2000

Name and mailing address of the ISA/US Commissioner of Patents and Trademarks Box PCT Washington, D.C. 20231 Facsimile No. (703) 305-3230	Authorized officer UYEN LE Telephone No. (703) 305-4134
---	---